DEDICATED MCU FAMILY

FOR TV/MONITOR APPLICATIONS

DATABOOK

1st EDITION

APRIL 1994

USE IN LIFE SUPPORT DEVICES OR SYSTEMS MUST BE EXPRESSLY AUTHORIZED

SGS-THOMSON PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF SGS-THOMSON Microelectronics. As used herein:

1. Life support devices or systems are those which (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided with the product, can be reasonably expected to result in significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can reasonably be expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

TABLE OF CONTENTS

INTRODUCTION

SGS-THOMSON's dedicated video product range is so extensive that it covers virtually all needs for the developers and producers of televisions, VCRs, monitors, satellite decoders and receivers and related applications. Devices for power and graphic circuits, and signal devices are found in the Video Products Databooks and Application Manuals. Complementing these products, and performing the central control functions and user interfaces of modern multistandard systems, SGS-THOMSON's dedicated microcontroller range for TV and Monitor has expanded to cover all model requirements, from low end TVs with Remote Control and On-Screen Display through Multi-Standard Monitors to high end TVs with Digital Data Slicers and expanded OSD capabilities for Closed Caption. As a result SGS-THOMSON has collected all these products into one book, as each microcontroller device is applicable for more than one area of use.

SGS-THOMSON's microcontrollers have two roots, the successful non-dedicated families for general applications, and secondly the company's capabilities in non-volatile memory technologies. Together with these, and with the experience brought from a long history of involvement in TV and Monitor applications, three families of dedicated microcontrollers have been developed, each with integrated features making them attractive and cost-effective system solutions.

These integrated functions bring features such as: on-chip EEPROM for parameter and channel storage; 14-bit Voltage Synthesizers on-chip for frequency tuning; On-Screen Displays (from 5 to 15 lines), also working together with Data Slicers for US Closed Caption; High current outputs for direct LED drive; inputs for immediate remote control commands; on-chip Digital to Analog converters for Analog controls and Sync Pulse Processing for determination of Video Standards, and a module for automatic East-West Pin Cushion Correction. Completing and adding to the dedicated functions is the flexibility that only a microcontroller-based system can offer.

For examples of full solutions based on microcontrollers, ask your local marketing for details of a series of complete chassis, with software (see table), fully available for evaluation of both the microcontroller functions and those of the dedicated power and signal products from SGS-THOMSON.

Standard Software Packages For Standard Chassis

SOFTWARE NAME	MCU	FS TUNING	VS TUNING	TELETEXT CHIP	TELETEXT BASIC	TELETEXT TOP	FLOF1.5/ LIST	X26	VIDEO PROCESSOR	OSD STYLE	AVAILABILITY CHASSIS	AVAILABILITY DOC.
BRIGITTE	ST6356	NO	YES	STV5345+ SAA5231	4Pgs	NO	NO	NO	TDA8217 PAL	SIMPLE	NOW	NOW
CRISTINA	ST6367	NO	YES	STV5345+ SAA5231	4Pgs	NO	NO	NO	STV2110A PAL/SECAM	SIMPLE	NOW	NOW
	ST6367	NO	YES	STV5345+ SAA5231	4Pgs	NO	NO	NO	STV2102A PAL	SIMPLE	NOW	NOW
DORA	ST6387	NO	YES	STV5345+ SAA5231	NO	NO	8Pgs	YES	STV2110A PAL/SECAM	SIMPLE	Q4/93	Q4/93

Consult your SGS-THOMSON marketing contact for an update on this chart

ST636x ROM Devices. MCU For TV Tuning with On-Screen Display

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	AFC	VS	D/A	COLOUR PINS	EMULATING DEVICES
ST6365	8K	256	384	YES	YES	4	3	ST63E85, ST63T85
ST6367	8K	256	384	YES	YES	6	3	ST63E87, ST63T87
ST6375	14K	256	384	YES	YES	4	3	ST63E85, ST63T85
ST6377	14K	256	384	YES	YES	6	3	ST63E87, ST63T87
ST6385	20K	256	384	YES	YES	4	3	ST63E85, ST63T85
ST6387	20K	256	384	YES	YES	6	3	ST63E87, ST63T87

ST63E6x EPROM/OTP Devices. MCU For TV Tuning with On-Screen Display

DEVICE	EPROM (Bytes)	OTPROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	AFC	VS	D/A	COLOUR PINS	TARGET ROM DEVICES
ST63E85	20K		256	384	YES	YES	4	3	ST6365, 75, 85
ST63T85		20K	256	384	YES	YES	4	3	ST6365, 75, 85
ST63E87	20K		256	384	YES	YES	6	3	ST6367, 77 87
ST63T87		20K	256	384	YES	YES	6	3	ST6367, 77, 87

ST631xx ROM Devices. MCU For TV Frequency and Voltage Synthesis with On-Screen Display

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	I/O	KBY I/O	BSW OUT	AFC	VS	D/A	PACK.	EMUL. DEVICES
ST63126	8K	256	128	12	3	4	YES	NO	4	PDIP40	ST63E126
ST63156	8K	256	128	11	3	4	YES	YES	4	PDIP40	ST63E156
ST63140	8K	256	128	6	3	3	YES	YES	1	PDIP28	ST63E140
ST63142	8K	256	128	6	3	3	YES	NO	1	PDIP28	ST63E142

ST631xx EPROM/OTP Devices. MCU For TV Frequency and Voltage Synthesis with On-Screen Display

DEVICE	EPROM (Bytes)	OTP ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	I/O	KBY I/O	BSW OUT	AFC	VS	D/A	PACK.	TARGET ROM DEVICES
ST63E140	8K		256	128	6	3	3	YES	YES	1	PDIP28	ST63140
ST63T140		8K	256	128	6	3	3	YES	YES	1	PDIP28	ST63140
ST63E142	8K		256	128	6	3	3	YES	NO	1	PDIP28	ST63142
ST631T42		8K	256	128	6	3	3	YES	NO	1	PDIP28	ST63142
ST63E126	8K		256	128	12	3	4	YES	NO	4	PDIP40	ST63126
ST63T126		8K	256	128	12	3	4	YES	NO	4	PDIP40	ST63126
ST63E156	8K		256	128	11	3	4	YES	YES	4	PDIP40	ST63156
ST63T156		8K	256	128	11	3	4	YES	YES	4	PDIP40	ST63156

ST6369 ROM Device. MCU For Digital Controlled Multi-Frequency Monitor

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	A/D	14-bit D/A	6-bit D/A	EMULATING DEVICES
ST6369	8K	256	384	1	1	6	ST63E69, ST63T69

ST6369 EPROM/OTP Device. MCU For Digital Controlled Multi-Frequency Monitor

DEVICE	EPROM (Bytes)	OTPROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	A/D	14-bit D/A	6-bit D/A	TARGET ROM DEVICE
ST63E69	8K		256	384	1	1	6	ST6369
ST63T69		8K	256	384	1	1	6	ST6369

ST7271 ROM Device. MCU For Digital Controlled Multi-Frequency Monitor

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	D/A Converter 12-bit	D/A Converter 10-bit	Sync. Proc.	Timer	Package	EMULATING DEVICES*
ST7271N5	16K	256	512	2	14	1	1	PSDIP56	ST72E71, ST72T71
ST7271J1	16K	256	512	2	10	1	1	PSDIP42	ST72E71, ST72T71

* Note. Contact SGS-THOMSON Marketing for further information.

ST929x ROM Devices. With On-Screen Display and SPI

DEVICE	ROM (Bytes)	RAM (Bytes)	I/O	A/D	D/A Converter 8-bit	D/A Converter 14-bit	Data Slicer	Timer	Package	EMULATING DEVICES*
ST9291N6	32K	640	42	1	8	1	-	2	PSDIP56	ST92E91N6
ST9291J6	32K	640	32	1	8	1	-	2	PSDIP42	ST92E91J6
ST9293J7	48K	768	31	1	-	-	-	2	PSDIP42	ST92E93J7
ST9294N6	32K	640	42	1	8	-	1	2	PSDIP56	ST92E91N6
ST9294J6	32K	640	31	1	8	-	1	2	PSDIP42	ST92E94J6

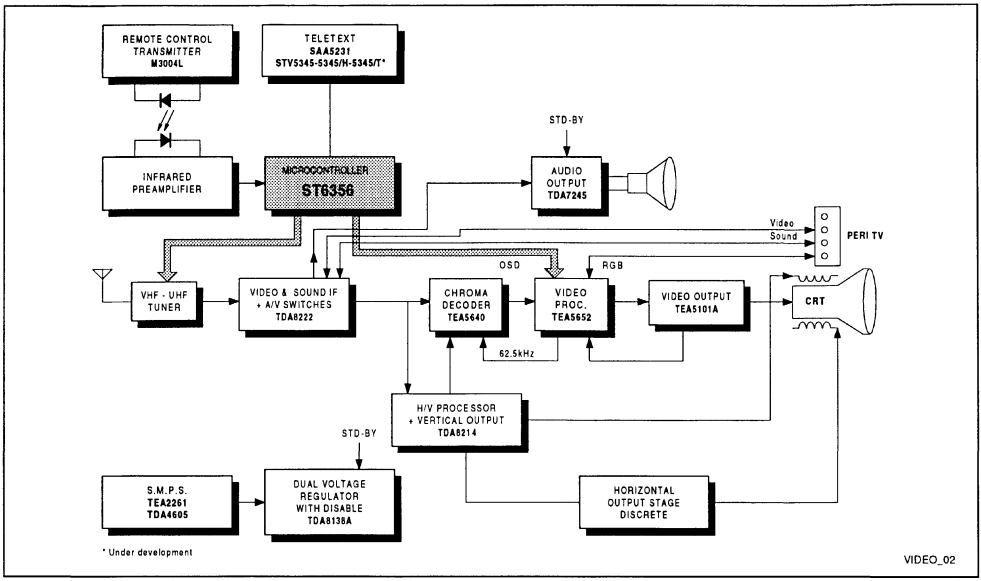
ST92E9x EPROM Devices. With On-Screen Display and SPI

DEVICE	EPROM (Bytes)	RAM (Bytes)	I/O	A/D	D/A Converter 8-bit	D/A Converter 14-bit	Data Slicer	Timer	Package	EMULATING DEVICES*
ST92E91N6	32K	640	42	1	8	1	-	2	CSDIP56W	ST9291N6
ST92E91J6	32K	640	32	1	8	1	-	2	CSDIP42W	ST9291J6
ST92E93J7	48K	768	31	1	-	-	-	2	CSDIP42W	ST9293J7
ST92E94N6	32K	640	42	1	8	-	1	2	CSDIP56W	ST9291N6
ST92E94J6	32K	640	31	1	8	-	1	2	CSDIP42W	ST9294J6

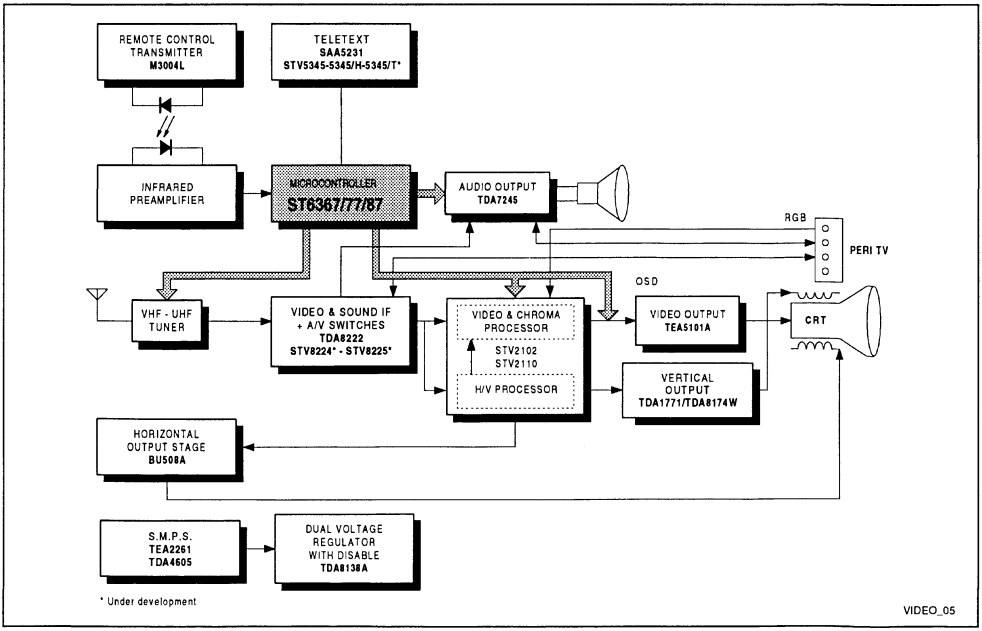
ST90R5x, R9x ROMless Devices. With Bankswitch and A/D Converter

DEVICE	ROM (Bytes)	RAM (Bytes)	I/O Port	A/D	Timer	SPI	SCI	Package
ST90R51	-	-	54	1	4	1	2	PQFP80
ST90R91C1	-	1536	35	1	3	1	-	PLCC68
ST90R91Q1	-	1536	40	1	3	1	-	PQFP80

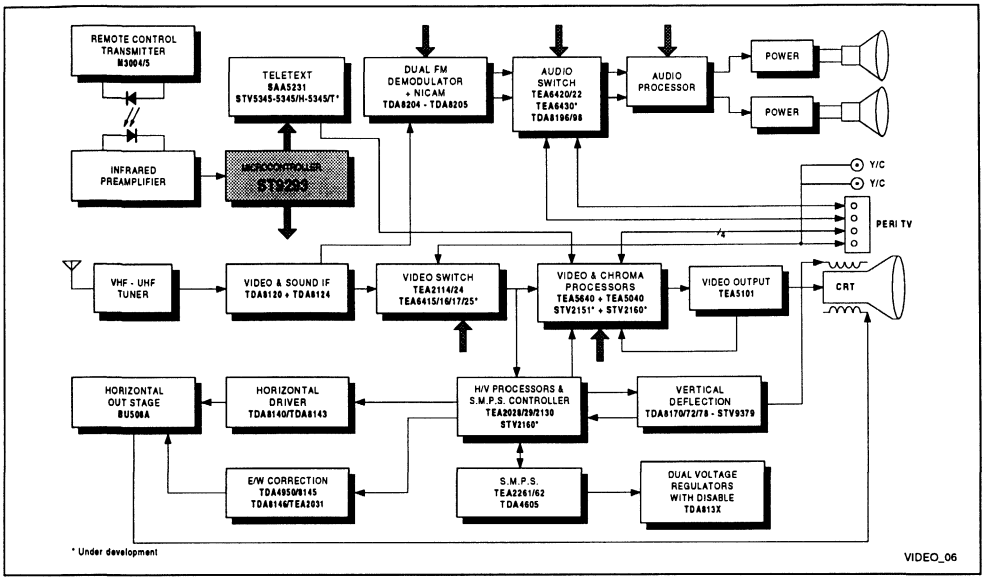
PAL/SECAM low cost chassis (BG/DK standard)



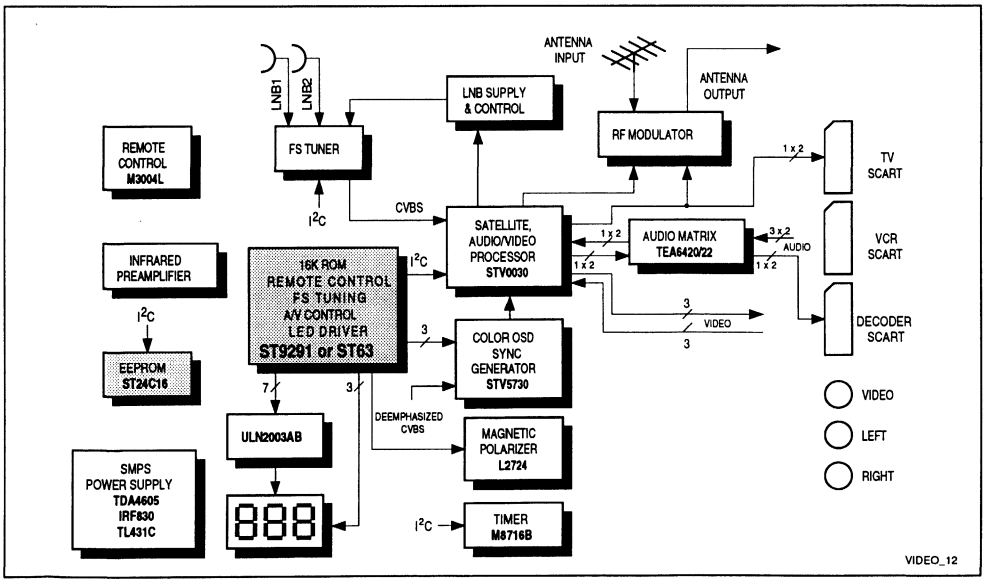
PAL or PAL/SECAM color TV set (BG/DK/L)



Flexible Multistandard CTV



Satellite video Receivers



GENERAL INDEX

8 Bit MCUs

8/16 Bit MCUs

8 Bit MCUs

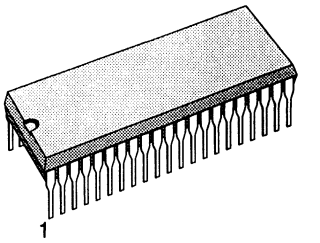
SGS-THOMSON MICROELECTRONICS

ST6365, ST6375, ST6385
ST6367, ST6377, ST6387

8-BIT HCMOS MCUs WITH ON-SCREEN DISPLAY FOR TV TUNING

PRELIMINARY DATA

- 4.5 to 6V supply operating range
- 8MHz Maximum Clock Frequency
- User Program ROM: up to 20140 bytes
- Reserved Test ROM: up to 340 bytes
- Data ROM: user selectable size
- Data RAM: 256 bytes
- Data EEPROM: 384 bytes
- 42-Pin Shrink Dual in Line Plastic Package
- Up to 22 software programmable general purpose Inputs/Outputs, including 2 direct LED driving Outputs
- Two Timers each including an 8-bit counter with a 7-bit programmable prescaler
- Digital Watchdog Function
- Serial Peripheral Interface (SPI) supporting S-BUS/ I^2C BUS and standard serial protocols
- SPI for external frequency synthesis tuning
- 14 bit counter for voltage synthesis tuning
- Up to Six 6-Bit PWM D/A Converters
- AFC A/D converter with 0.5V resolution
- Five interrupt vectors (IRIN/NMI, Timer 1 & 2, VSYNC, PWR INT.)
- On-chip clock oscillator
- 5 Lines by 15 Characters On-Screen Display Generator with 128 Characters
- All ROM types are supported by pin-to-pin EPROM, and OTP versions.
- The development tool of the ST636x,7x,8x microcontrollers consists of the ST638x-EMU emulation and development system to be connected via a standard RS232 serial line to an MS-DOS Personal Computer.



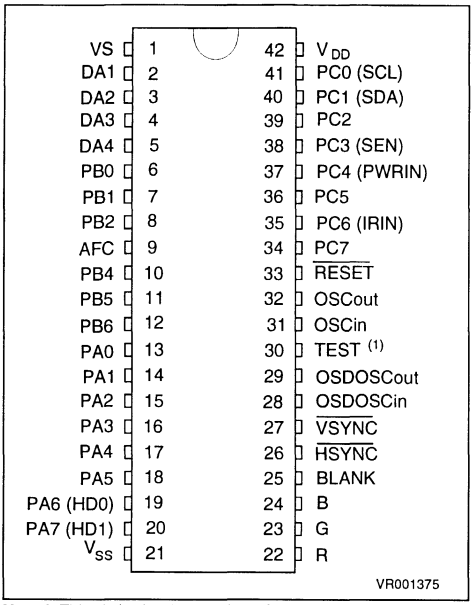
PSDIP42

(Ordering Information at the end of the datasheet)

DEVICE SUMMARY

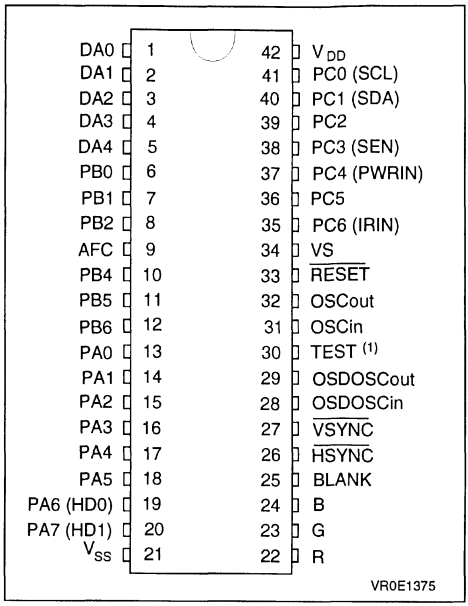
DEVICE	ROM (Bytes)	D/A Converter
ST6365	8K	4
ST6367	8K	6
ST6375	14K	4
ST6377	14K	6
ST6385	20K	4
ST6387	20K	6

Figure 2. ST6365, 75, 85 Pin Configuration



Note 1. This pin is also the VPP input for EPROM based devices

Figure 1. ST6367, 77, 87 Pin Configuration



GENERAL DESCRIPTION

The ST6365,67,75,77,85,87 microcontrollers are members of the 8-bit HCMOS ST638x family, a series of devices specially oriented to TV applications. Different ROM size and peripheral configurations are available to give the maximum application and cost flexibility. All ST638x members are based on a building block approach: a common core is surrounded by a combination of on-chip peripherals (macrocells) available from a standard library. These peripherals are designed with the same Core technology providing full compatibility and short design time. Many of these macrocells are specially dedicated to TV applications. The macrocells of the ST638x family are: two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), a digital hardware activated watchdog function (DHWD), a 14-bit voltage synthesis tuning peripheral, a Serial Peripheral Interface (SPI), up to six 6-bit PWM D/A converters, an AFC A/D converter with 0.5V resolution, an on-screen display (OSD) with 15 characters per line and 128 characters (in two banks each of 64 characters). In addition the following memory resources are available: program ROM (up to 20K), data RAM (256 bytes), EEPROM (384 bytes). Refer to pin configurations figures and to ST638x device summary (Table 1) for the definition of ST638x family members and a summary of differences among the different types.

Figure 3. ST6365,67,75,77,85,87 Block Diagram

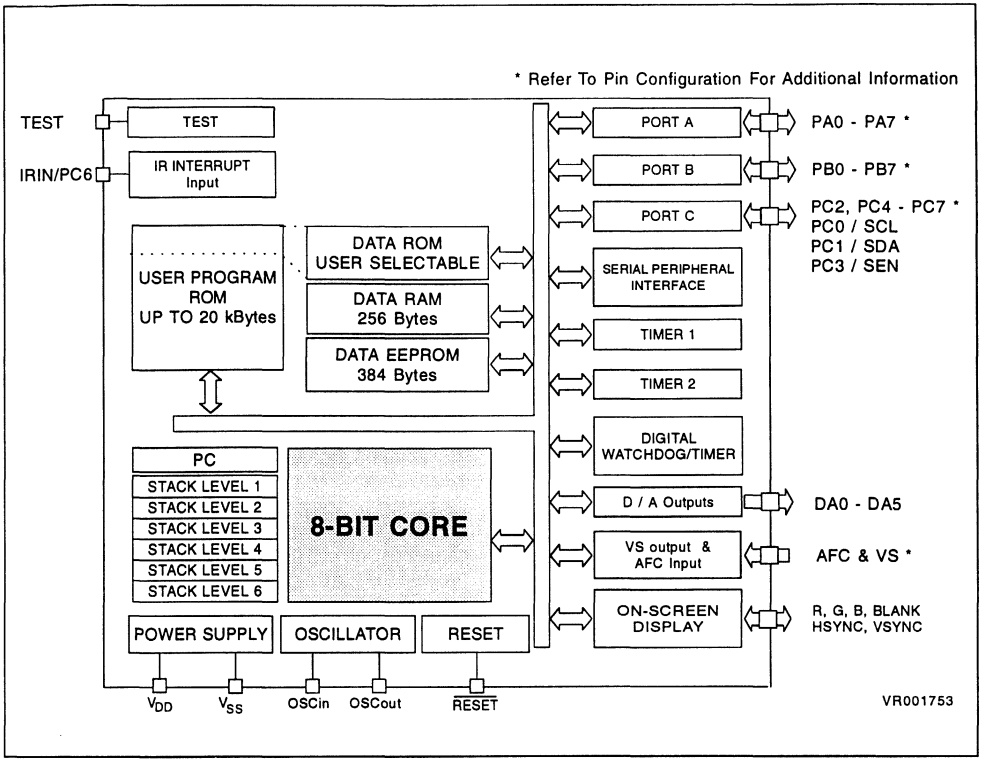


Table 1. Device Summary

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	AFC	VS	D/A	COLOUR PINS	EMULATING DEVICES
ST6365	8K	256	384	YES	YES	4	3	ST63P85, ST63E85
ST6367	8K	256	384	YES	YES	6	3	ST63P87, ST63E87
ST6375	14K	256	384	YES	YES	4	3	ST63P85, ST63E85
ST6377	14K	256	384	YES	YES	6	3	ST63P87, ST63E87
ST6385	20K	256	384	YES	YES	4	3	ST63P85, ST63E85
ST6387	20K	256	384	YES	YES	6	3	ST63P87, ST63E87

PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCin, OSCout. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to start the microcontroller to the beginning of its program. Additionally the quartz crystal oscillator will be disabled when the $\overline{\text{RESET}}$ pin is low to reduce power consumption during reset phase.

TEST. The TEST pin must be held at V_{SS} for normal operation.

PA0-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured as either an input with or without pull-up resistor or as an output under software control of the data direction register. Pins PA4 to PA7 are configured as open-drain outputs (12V drive). On PA4-PA7 pins the input pull-up option is not available while PA6 and PA7 have additional current driving capability (25mA, V_{OL}:1V). PA0 to PA3 pins are configured as push-pull.

PB0-PB2, PB4-PB6. These 6 lines are organized as one I/O port (B). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register.

PC0-PC7. These 8 lines are organized as one I/O port (C). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register. Pins PC0 to PC3 are configured as open-drain (5V drive) in output mode while PC4 to PC7 are open-drain with 12V drive and the input pull-up options does not exist on these four pins. PC0, PC1 and PC3 lines when in output mode are "ANDed" with the SPI control signals and are all open-drain. PC0 is connected to the SPI clock signal (SCL), PC1 with the SPI data signal (SDA) while PC3 is connected with SPI enable signal (SEN, used in S-BUS protocol). Pin PC4 and PC6 can also be inputs to software programmable edge sensitive latches which can generate interrupts; PC4 can be connected to Power Interrupt while PC6 can be connected to the IRIN/NMI interrupt line.

DA0-DA5. These pins are the six PWM D/A outputs of the 6-bit on-chip D/A converters. These lines have open-drain outputs with 12V drive. The output repetition rate is 31.25KHz (with 8MHz clock).

AFC. This is the input of the on-chip 10 levels comparator that can be used to implement the AFC function. This pin is an high impedance input able to withstand signals with a peak amplitude up to 12V.

OSDOSCin, OSDOSCout. These are the On Screen Display oscillator terminals. An oscillation capacitor and coil network have to be connected to provide the right signal to the OSD.

HSYNC, VSYNC. These are the horizontal and vertical synchronization pins. The active polarity of these pins to the OSD macrocell can be selected by the user as ROM mask option. If the device is specified to have negative logic inputs, then these signals are low the OSD oscillator stops. If the device is specified to have positive logic inputs, then when these signals are high the OSD oscillator stops. VSYNC is also connected to the VSYNC interrupt.

R, G, B, BLANK. Outputs from the OSD. R, G and B are the color outputs while BLANK is the blanking output. All outputs are push-pull. The active polarity of these pins can be selected by the user as ROM mask option.

VS. This is the output pin of the on-chip 14-bit voltage synthesis tuning cell (VS). The tuning signal present at this pin gives an approximate resolution of 40KHz per step over the UHF band. This line is a push-pull output with standard drive.

Table 2. Pin Summary

Pin Function	Description
DA0 to DA5	Output, Open-Drain, 12V
AFC	Input, High Impedance, 12V
VS	Output, Push-Pull
R,G,B, BLANK	Output, Push-Pull
HSYNC, VSYNC	Input, Pull-up, Schmitt Trigger
OSDOSCin	Input, High Impedance
OSDOSCout	Output, Push-Pull
TEST	Input, Pull-Down
OSCin	Input, Resistive Bias, Schmitt Trigger to Reset Logic Only
OSCout	Output, Push-Pull
RESET	Input, Pull-up, Schmitt Trigger Input
PA0-PA3	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PA4-PA5	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
PA6-PA7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input, High Drive
PB0-PB2	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PB4-PB6	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PC0-PC3	I/O, Open-Drain, 5V , Software Input Pull-up, Schmitt Trigger Input
PC4-PC7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
V_{DD}, V_{SS}	Power Supply Pins

ST638x CORE

The Core of the ST638x Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal addresses, data, and control busses. The in-core communication is arranged as shown in the following block diagram figure; the controller being externally linked to both the reset and the oscillator, while the core is linked to the dedicated on-chip macrocells peripherals via the serial data bus and indirectly for interrupt purposes through the control registers.

Registers

The ST638x Family Core has five registers and three pairs of flags available to the programmer. They are shown in Figure 5 and are explained in the following paragraphs together with the program and data memory page registers.

Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator is addressed in the data space as RAM location at the FFh address.
Accordingly, the ST638x instruction set can use the accumulator as any other register of the data space.

Figure 5. ST638x Core Programming Model

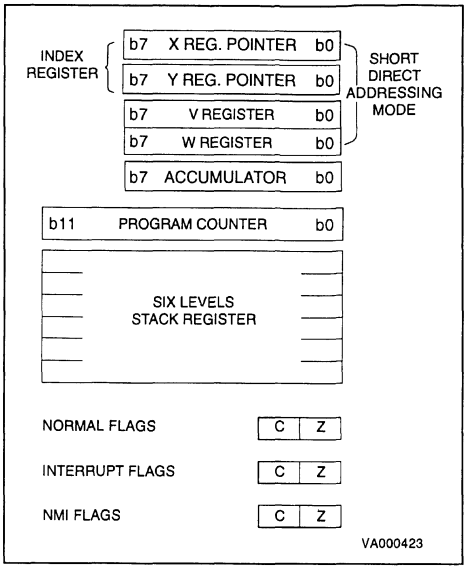
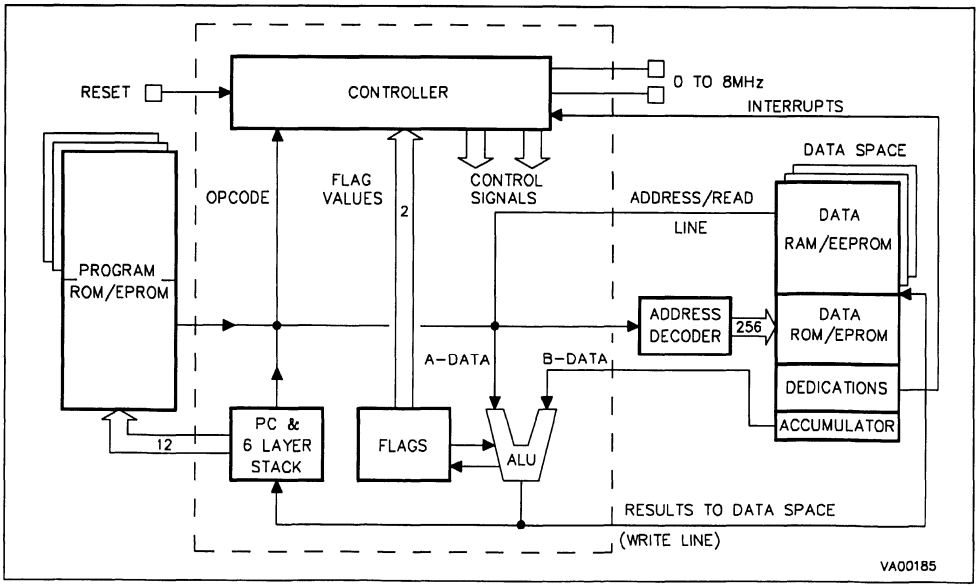


Figure 4. ST6365,67,75,77,85,87 Core Block Diagram



ST638x CORE (Continued)

Indirect Registers (X, Y). These two indirect registers are used as pointers to the memory locations in the data space. They are used in the register-indirect addressing mode.These registers can be addressed in the data space as RAM locations at the 80h (X) and 81h (Y) addresses. They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST638x instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save one byte in short direct addressing mode. These registers can be addressed in the data space as RAM locations at the 82h (V) and 83h (W) addresses. They can also be accessed with the direct and bit direct addressing modes. Accordingly, the ST638x instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC)

The program counter is a 12-bit register that contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or an address of operand. The 12-bit length allows the direct addressing of 4096 bytes in the program space. Nevertheless, if the program space contains more than 4096 locations, the further program space can be addressed by using the Program ROM Page Register. The PC value is incremented, after it is read for the address of the current instruction, by sending it through the ALU, so giving the address of the next byte in the program. To execute relative jumps the PC and the offset values are shifted through the ALU, where they will be added, and the result is shifted back into the PC. The program counter can be changed in the following ways:

JP (Jump) instruction.... PC= Jump address

CALL instruction..... PC= Call address

Relative Branch instructions..... PC= PC+offset

Interrupt..... PC= Interrupt vector

Reset..... PC= Reset vector

RET & RETI instructions..... PC= Pop (stack)

Normal instruction ..... PC= PC+1

Flags (C, Z)

The ST638x Core includes three pairs of flags that correspond to 3 different modes: normal mode, interrupt mode and Non-Maskable-Interrupt-Mode. Each pair consists of a CARRY flag and a ZERO flag. One pair (CN, ZN) is used during normal operation, one pair is used during the interrupt mode (CI,ZI) and one is used during the not-maskable interrupt mode (CNMI, ZNMI).

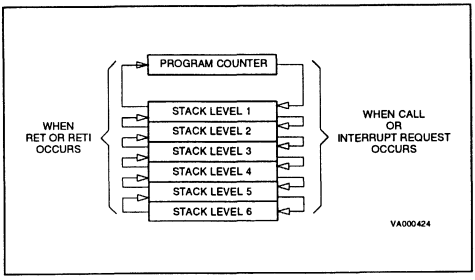
The ST638x Core uses the pair of flags that corresponds to the actual mode: as soon as an interrupt (resp. a Non-Maskable-Interrupt) is generated, the ST638x Core uses the interrupt flags (resp. the NMI flags) instead of the normal flags. When the RETI instruction is executed, the normal flags (resp. the interrupt flags) are restored if the MCU was in the normal mode (resp. in the interrupt mode) before the interrupt. Should be observed that each flag set can only be addressed in its own routine (Not-maskable interrupt, normal interrupt or main routine). The interrupt flags are not cleared during the context switching and so, they remain in the state they were at the exit of the last routine switching.

The Carry flag is set when a carry or a borrow occurs during arithmetic operations, otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction, and participates in the rotate left instruction.

The Zero flag is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

The switching between these three sets is automatically performed when an NMI, an interrupt and a RETI instructions occur. As the NMI mode is automatically selected after the reset of the MCU, the ST638x Core uses at first the NMI flags.

Figure 6. Stack Operation



ST638x CORE (Continued)

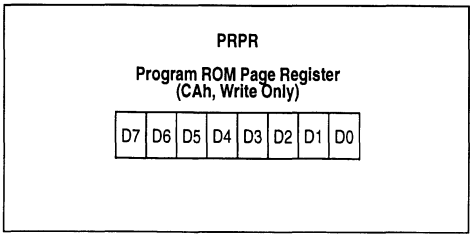
Stack

The ST638x Core includes true LIFO hardware stack that eliminates the need for a stack pointer. The stack consists of six separate 12-bit RAM locations that do not belong to the data space RAM area. When a subroutine call (or interrupt request) occurs, the contents of each level is shifted into the next level while the content of the PC is shifted into the first level (the value of the sixth level will be lost). When subroutine or interrupt return occurs (RET or RETI instructions), the first level register is shifted back into the PC andthe value of each level is shifted back into the previous level. These two operating modes are described in Figure 6. Since the accumulator, as all other data space registers, is not stored in this stack the handling of this registers shall be performed inside the subroutine. The stack pointer will remain in its deepest position, if more than 6 calls or interrupts are executed, so that the last return address will be lost. It will also remain in its highest position if the stack is empty and a RET or RETI is executed. In this case the next instruction will be executed.

Memory Registers

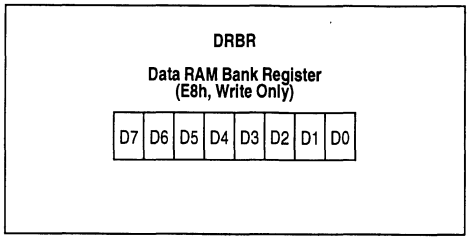
The PRPR can be addressed like a RAM location in the Data Space at the CAh address; nevertheless it is a write-only register that can not be accessed with single-bit operations. This register is used to select the 2-Kbyte ROM bank of the Program Space that will be addressed. The number of the page has to be loaded in the PRPR. The PRPR is not cleared during the MCU initialization and should therefore be defined before jumping out of the static page. Refer to the Program Space description for additional information concerning the use of this register. The PRPR is not modified when an interrupt or a subroutine occurs.

Figure 7. Program ROM Page Register



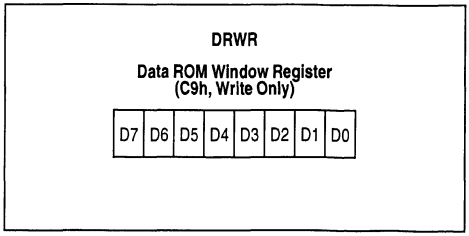
The DRBR can be addressed like a RAM location in the Data Space at the E8h address, nevertheless it is write-only register that can not be accessed with single-bit operations. This register is used to select the desired 64-byte RAM/EEPROM bank of the Data Space. The number of the bank has to be loaded in the DRBR and the instruction has to point to the selected location as it was in the 0 bank (from 00h address to 3Fh address). This register is undefined after Reset. Refer to the Data Space description for additional information. The DRBR register is not modified when a interrupt or a subroutine occurs.

Figure 8. Data RAM Bank Register



The DRWR register can be addressed like a RAM location in the Data Space at the C9h address, nevertheless it is write-only register that can not be accessed with single-bit operations. This register is used to move up and down the 64-byte read-only data window (from the 40h address to 7Fh address of the Data Space) along the ROM of the MCU by step of 64 bytes. The effective address of the byte to be read as a data in the ROM is obtained by the concatenation of the 6 less significant bits of the address given in the instruction (as less significant bits) and the content of the DRWR (as most significant bits). Refer to the Data Space description for additional information.

Figure 9. Data ROM Window Register



MEMORY SPACES

The MCUs operate in three different memory spaces: Stack Space, Program Space, and Data Space. (Figure 10 refers to the ST638x which has a total of 20K bytes of ROM).

Stack Space

The stack space consists of six 12 bit registers that are used for stacking subroutine and interrupt return addresses plus the current program counter register.

Program Space

The program space is physically implemented in the ROM and includes all the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, the reserved test area and user vectors. It is addressed thanks to the 12-bit Program Counter register (PC register) and so, the ST638x Core can directly address up to 4K bytes of Program Space. Nevertheless, the Program Space can be extended by the addition of 2-Kbyte ROM banks as it is shown in Figure 12 in which a 20K bytes memory is described. These banks are addressed by pointing to the 000h-7FFh locations of the Program Space thanks to the Program Counter, and by writing the appropriate code in the Program ROM Page Register (PRPR) located at the CAh address of the Data Space. Because interrupts and common subroutines should be available all the time only the lower 2K byte of the 4K program space are bank switched while the upper 2K byte can be seen as static space. Table 3 gives the different codes that allows the selection of the corresponding banks. Note that, from the memory point of view, the Page 1 and the Static Page represent the same physical memory: it is only a different way of addressing the same location. On the ST6385 and ST6387, a total of 20480 bytes of ROM have been implemented; 20140 are available as user ROM while 340 are reserved for testing.

Figure 11. ST638x 20K Bytes Program Space Addressing Description

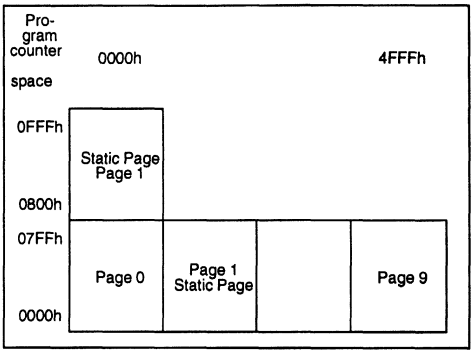
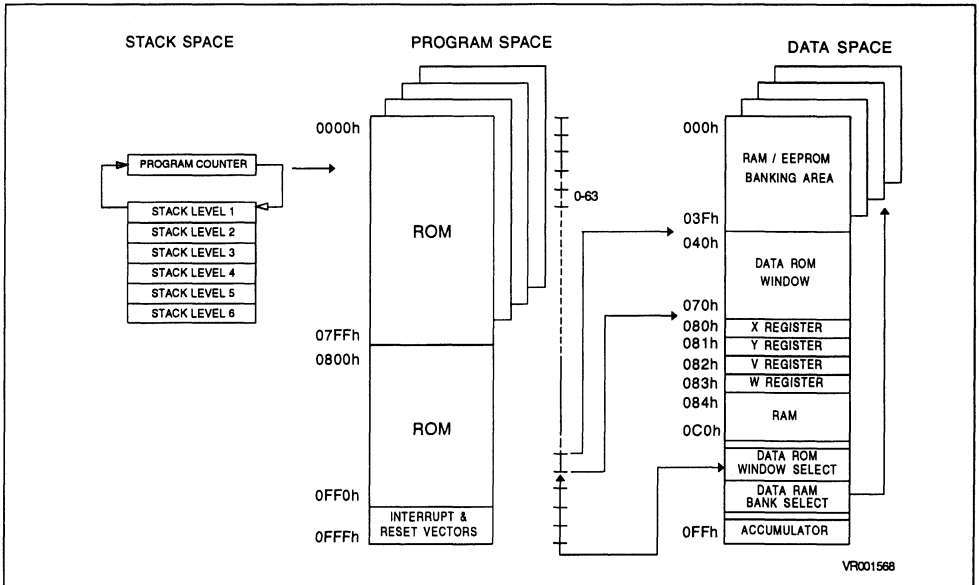
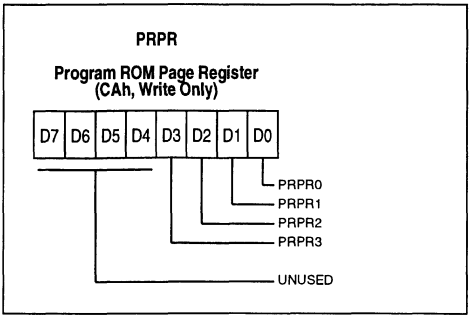


Figure 10. ST638x Memory Addressing Description Diagram



MEMORY SPACES (Continued)

Figure 12. Program ROM Page Register



D7-D5. These bits are not used.

PRPR4-PRPR0. These are the program ROM banking bits and the value loaded selects the corresponding page to be addressed in the lower part of 4K program address space as specified in Table 3. This register is undefined on reset.

Note. The number of bits implemented depends on the size of the ROM of the device. Only the lower part of address space has been bank-switched because interrupt vectors and common subroutines should be available all the time. The reason of this structure is due to the fact that it is not possible to jump from a dynamic page to another, unless jumping back to the static page, changing contents of PRPR, and, than, jumping to a different dynamic page.

Care is required when handling the PRPR as it is write only. For this reason, it is not allowed to change the PRPR contents while executing interrupts drivers, as the driver cannot save and than restore its previous content. Anyway, this operation may be necessary if the sum of common routines and interrupt drivers will take more than 2K bytes; in this case could be necessary to divide the interrupt driver in a (minor) part in the static page (start and end), and in the second (major) part in one dynamic page. If it is impossible to avoid the writing of this register in interrupts drivers, an image of this register must be saved in a RAM location, and each time the program writes the PRPR it writes also the image register. The image register must be written first, so if an interrupt occurs between the two instructions the PRPR is not affected.

Table 3. ST636x, 7x, 8x Program ROM Page Register Coding

PRPR3	PRPR2	PRPR1	PRPR0	PC11	Memory Page
X	X	X	X	1	Static Page (Page 1)
0	0	0	0	0	Page 0
0	0	0	1	0	Page 1 (Static Page)
0	0	1	0	0	Page 2
0	0	1	1	0	Page 3
0	1	0	0	0	Page 4
0	1	0	1	0	Page 5
0	1	1	0	0	Page 6
0	1	1	1	0	Page 7
1	0	0	0	0	Page 8
1	0	0	1	0	Page 9

MEMORY SPACES (Continued)

Table 4. ST638x Program ROM Map (up to 20K Bytes)

ROM Page	Device Address	Description
PAGE 0	0000h-007Fh 0080h-07FFh	Reserved User ROM
PAGE 1 "STATIC"	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
PAGE 2	0000h-000Fh 0010h-07FFh	Reserved User ROM
PAGE 3	0000h-000Fh 0010h-07FFh	Reserved User ROM (End of 8k ST6365, 67)
PAGE 4	0000h-000Fh 0010h-07FFh	Reserved User ROM
PAGE 5	0000h-000Fh 0010h-07FFh	Reserved User ROM
PAGE 6	0000h-000Fh 0010h-07FFh	Reserved User ROM (End of 14k ST6375, 77)
PAGE 7	0000h-000Fh 0010h-07FFh	Reserved User ROM
PAGE 8	0000h-000Fh 0010h-07FFh	Reserved User ROM
PAGE 9	0000h-000Fh 0010h-07FFh	Reserved User ROM (End of 20k ST6385, 87)

MEMORY SPACES (Continued)

Data Space

The instruction set of the ST638x Core operates on a specific space, named Data Space that contains all the data necessary for the processing of the program. The Data Space allows the addressing of RAM (256 bytes for the ST638x family), EEPROM (384 bytes), ST638x Core/peripheral registers, and read-only data such as constants and the look-up tables.

Figure 13. ST638x Data Space

b7 b0	
DATA RAM/EEPROM/OSD BANK AREA	000h 03Fh
DATA ROM WINDOW AREA	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
PORT C DATA REGISTER	0C2h
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
PORT C DIRECTION REGISTER	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h
DATA ROM WINDOW REGISTER	0C9h
PROGRAM ROM PAGE REGISTER	0CAh
RESERVED	0CBh
SPI DATA REGISTER	0CCh
RESERVED	0CDh 0D1h
TIMER 1 PRESCALER REGISTER	0D2h
TIMER 1 COUNTER REGISTER	0D3h
TIMER 1 STATUS/CONTROL REG.	0D4h
RESERVED	0D5h 0D7h
WATCHDOG REGISTER	0D8h

Figure 14. ST638x Data Space (Continued)

b7 b0	
RESERVED	0D9h
TIMER 2 PRESCALER REGISTER	0DAh
TIMER 2 COUNTER REGISTER	0DBh
TIMER 2 STATUS CONTROL REG.	0DCh
RESERVED	0DDh 0DFh
DA0 DATA/CONTROL REGISTER	0E0h
DA1 DATA/CONTROL REGISTER	0E1h
DA2 DATA/CONTROL REGISTER	0E2h
DA3 DATA/CONTROL REGISTER	0E3h
AFC, IR & OSD RESULT REGISTER	0E4h
OUTPUTS CONTROL REGISTER	0E5h
DA4 DATA/CONTROL REGISTER	0E6h
DA5 DATA/CONTROL REGISTER	0E7h
DATA RAM BANK REGISTER	0E8h
DEDIC. LATCHES CONTROL REG.	0E9h
EEPROM CONTROL REGISTER	0EAh
SPI CONTROL REGISTER 1	0EBh
SPI CONTROL REGISTER 2	0ECh
OSD CHARAC. BANK SELECT REG.	0EDh
VS DATA REGISTER 1	0EEh
VS DATA REGISTER 2	0EFh
RESERVED	0F0h 0FEh
ACCUMULATOR	0FFh

OSD CONTROL REGISTERS LOCATED IN PAGE 6 OF BANKED DATA RAM

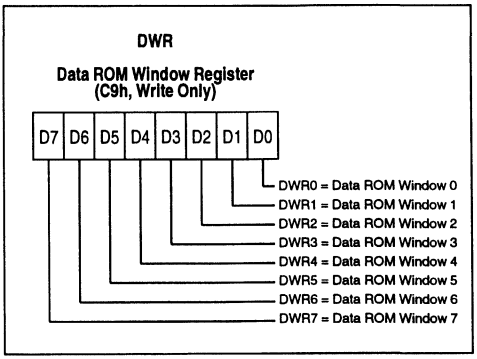
VERTICAL START ADDRESS REG.	010h
HORIZONTAL START ADDRESS REG.	011h
VERTICAL SPACE REGISTER	012h
HORIZONTAL SPACE REGISTER	013h
BACKGROUND COLOUR REGISTER	014h
GLOBAL ENABLE REGISTER	017h

MEMORY SPACES (Continued)

Data ROM Addressing. All the read-only data are physically implemented in the ROM in which the Program Space is also implemented. The ROM therefore contains the program to be executed and also the constants and the look-up tables needed for the program. The locations of Data Space in which the different constants and look-up tables are addressed by the ST638x Core can be considered as being a 64-byte window through which it is possible to access to the read-only data stored in the ROM. This window is located from the 40h address to the 7Fh address in the Data space and allows the direct reading of the bytes from the 000h address to the 03Fh address in the ROM. All the bytes of the ROM can be used to store either instructions or read-only data. Indeed, the window can be moved by step of 64 bytes along the ROM in writing the appropriate code in the Write-only Data ROM Window register (DRWR, location C9h). The effective address of the byte to be read as a data in the ROM is obtained by the concatenation of the 6 less significant bits of the address in the Data Space (as less significant bits) and the content of the DRWR (as most significant bits). So when addressing location 40h of data space, and 0 is loaded in the DRWR, the physical addressed location in ROM is 00h.

Note. The data ROM window cannot address windows above the 16k byte range.

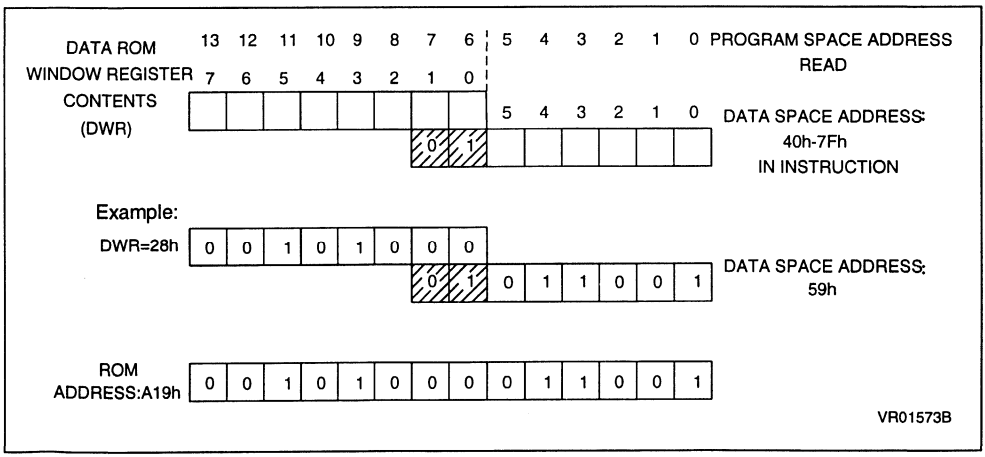
Figure 15. Data ROM Window Register



DWR7-DWR0. These are the Data Rom Window bits that correspond to the upper bits of data ROM program space. This register is undefined after reset.

Note. Care is required when handling the DRWR as it is write only. For this reason, it is not allowed to change the DRWR contents while executing interrupts drivers, as the driver cannot save and than restore its previous content. If it is impossible to avoid the writing of this register in interrupts drivers, an image of this register must be saved in a RAM location, and each time the program writes the DRWR it writes also the image register. The image register must be written first, so if an interrupt occurs between the two instructions the DRWR register is not affected.

Figure 16. Data ROM Window Memory Addressing

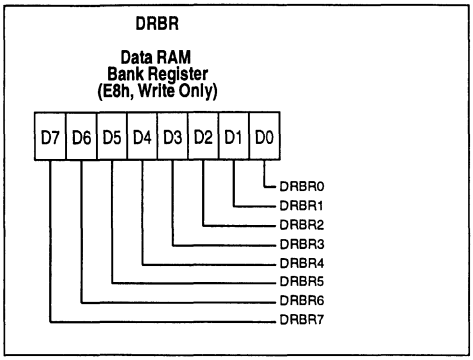


MEMORY SPACES (Continued)

Data RAM/EEPROM/OSD RAM Addressing

In all members of the ST638x family 64 bytes of data RAM are directly addressable in the data space from 80h to BFh addresses. The additional 192 bytes of RAM, the 384 bytes of EEPROM , and the OSD RAM can be addressed using the banks of 64 bytes located between addresses 00h and 3Fh. The selection of the bank is done by programming the Data RAM Bank Register (DRBR) located at the E8h address of the Data Space. In this way each bank of RAM, EEPROM or OSD RAM can select 64 bytes at a time. No more than one bank should be set at a time.

Figure 17. Data RAM Bank Register



DRBR7,DRBR1,DRBR0. These bits select the EEPROM pages.

DRBR6, DRBR5. Each of these bits, when set, will select one OSD RAM register page.

DRBR4,DRBR3,DRBR2. Each of these bits, when set, will select one RAM page.

This register is undefined after reset.

Table 5 summarizes how to set the Data RAM Bank Register in order to select the various banks or pages.

Note :
Care is required when handling the DRBR as it is write only. For this reason, it is not allowed to change the DRBR contents while executing interrupts drivers, as the driver cannot save and than restore its previous content. If it is impossible to avoid the writing of this register in interrupts drivers, an image of this register must be saved in a RAM location, and each time the program writes the DRBR it writes also the image register.
The image register must be written first, so if an interrupt occurs between the two instructions the DRBR is not affected.

Table 5. Data RAM Bank Register Set-up

DRBR Value		Selection
Hex.	Binary	
01h	0000 0001	EEPROM Page 0
02h	0000 0010	EEPROM Page 1
03h	0000 0011	EEPROM Page 2
81h	1000 0001	EEPROM Page 3
82h	1000 0010	EEPROM Page 4
83h	1000 0011	EEPROM Page 5
04h	0000 0100	RAM Page 2
08h	0000 1000	RAM Page 3
10h	0001 0000	RAM Page 4
20h	0010 0000	OSD Page 5
40h	0100 0000	OSD Page 6

MEMORY SPACES (Continued)

EEPROM Description

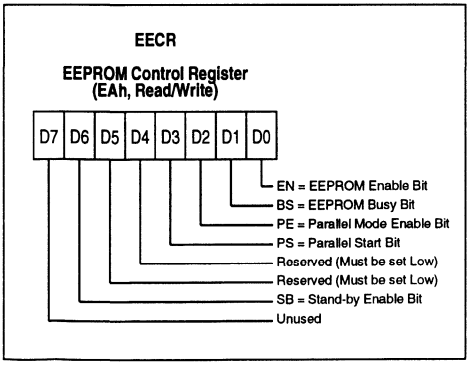
The data space of ST638x family from 00h to 3Fh is paged as described in Table 5. 384 bytes of EEPROM located in six pages of 64 bytes (pages 0,1,2,3,4 and 5, see Table 5).

Through the programming of the Data RAM Bank Register (DRBR=E8h) the user can select the bank or page leaving unaffected the way to address the static registers. The way to address the "dynamic" page is to set the DRBR as described in Table 5 (e.g. to select EEPROM page 0, the DRBR has to be loaded with content 01h, see Data RAM/EEPROM/OSD RAM addressing for additional information). Bits 0, 1 and 7 of the DRBR are dedicated to the EEPROM.

The EEPROM pages do not require dedicated instructions to be accessed in reading or writing. The EEPROM is controlled by the EEPROM Control Register (EECR=EAh). Any EEPROM location can be read just like any other data location, also in terms of access time.

To write an EEPROM location takes an average time of 5 ms (10ms max) and during this time the EEPROM is not accessible by the Core. A busy flag can be read by the Core to know the EEPROM status before trying any access. In writing the EEPROM can work in two modes: Byte Mode (BMODE) and Parallel Mode (PMODE). The BMODE is the normal way to use the EEPROM and consists in accessing one byte at a time. The PMODE consists in accessing 8 bytes per time.

Figure 18. EEPROM Control Register



D7. Not used

SB. *WRITE ONLY.* If this bit is set the EEPROM is disabled (any access will be meaningless) and the power consumption of the EEPROM is reduced to the leakage values.

D5, D4. Reserved for testing purposes, they must be set to zero.

PS. *SET ONLY.* Once in Parallel Mode, as soon as the user software sets the PS bit the parallel writing of the 8 adjacent registers will start. PS is internally reset at the end of the programming procedure. Note that less than 8 bytes can be written; after parallel programming the remaining undefined bytes will have no particular content.

PE. *WRITE ONLY.* This bit must be set by the user program in order to perform parallel programming (more bytes per time). If PE is set and the "parallel start bit" (PS) is low, up to 8 adjacent bytes can be written at the maximum speed, the content being stored in volatile registers. These 8 adjacent bytes can be considered as row, whose A7, A6, A5, A4, A3 are fixed while A2, A1 and A0 are the changing bytes. PE is automatically reset at the end of any parallel programming procedure. PE can be reset by the user software before starting the programming procedure, leaving unchanged the EEPROM registers.

BS. *READ ONLY.* This bit will be automatically set by the CORE when the user program modifies an EEPROM register. The user program has to test it before any read or write EEPROM operation; any attempt to access the EEPROM while "busy bit" is set will be aborted and the writing procedure in progress completed.

EN. *WRITE ONLY.* This bit MUST be set to one in order to write any EEPROM register. If the user program will attempt to write the EEPROM when EN= "0" the involved registers will be unaffected and the "busy bit" will not be set.

After RESET the content of EECR register will be 00h.

Notes :

When the EEPROM is busy (BS="1") the EECR can not be accessed in write mode, it is only possible to read BS status. This implies that as long as the EEPROM is busy it is not possible to change the status of the EEPROM control register. EECR bits 4 and 5 are reserved for test purposes, and must never be set to "1".

MEMORY SPACES (Continued)

Additional Notes on Parallel Mode. If the user wants to perform a parallel programming the first action should be the set to one the PE bit; from this moment the first time the EEPROM will be addressed in writing, the ROW address will be latched and it will be possible to change it only at the end of the programming procedure or by resetting PE without programming the EEPROM. After the ROW address latching the Core can "see" just one EEPROM row (the selected one) and any attempt to write or read other rows will produce errors. Do not read the EEPROM while PE is set.

As soon as PE bit is set, the 8 volatile ROW latches are cleared. From this moment the user can load data in the whole ROW or just in a subset. PS setting will modify the EEPROM registers corresponding to the ROW latches accessed after PE. For example, if the software sets PE and accesses EEPROM in writing at addresses 18h,1Ah,1Bh and then sets PS, these three registers will be modified at the same time; the remaining bytes will have no particular content. Note that PE is internally reset at the end of the programming procedure. This implies that the user must set PE bit between two parallel programming procedures. Anyway the user can set and then reset PE without performing any EEPROM programming. PS is a set only bit and is internally reset at the end of the programming procedure. Note that if the user tries to set PS while PE is not set there will not be any programming procedure and the PS bit will be unaffected. Consequently PS bit can not be set if EN is low. PS can be affected by the user set if, and only if, EN and PE bits are also set to one.

INTERRUPT

The ST638x Core can manage 4 different maskable interrupt sources, plus one non-maskable interrupt source (top priority level interrupt). Each source is associated with a particular interrupt vector that contains a Jump instruction to the related interrupt service routine. Each vector is located in the Program Space at a particular address (see Table 6). When a source provides an interrupt request, and the request processing is also enabled by the ST638x Core, then the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction). Finally, the PC is loaded with the address of the Jump instruction and the interrupt routine is processed.

The relationship between vector and source and the associated priority is hardware fixed for the different ST638x devices. For some interrupt sources it is also possible to select by software the kind of event that will generate the interrupt.

All interrupts can be disabled by writing to the GEN bit (global interrupt enable) of the interrupt option register (address C8h). After a reset, ST638x is in non maskable interrupt mode, so no interrupts will be accepted and NMI flags will be used, until a RETI instruction is executed. If an interrupt is executed, one special cycle is made by the core, during that the PC is set to the related interrupt vector address. A jump instruction at this address has to redirect program execution to the beginning of the related interrupt routine. The interrupt detecting cycle, also resets the related interrupt flag (not available to the user), so that another interrupt can be stored for this current vector, while its driver is under execution.

If additional interrupts arrive from the same source, they will be lost. NMI can interrupt other interrupt routines at any time, while other interrupts cannot interrupt each other. If more than one interrupt is waiting for service, they are executed according to their priority. The lower the number, the higher the priority. Priority is, therefore, fixed. Interrupts are checked during the last cycle of an instruction (RETI included). Level sensitive interrupts have to be valid during this period.

INTERRUPT (Continued)

Table 6. Interrupt Vectors/Sources Relationships

Interrupt Source	Associated Vector	Vector Address
PC6/IRIN Pin (1)	Interrupt Vector # 0 (NMI)	0FFCh-0FFDh
Timer 2	Interrupt Vector # 1	0FF6h-0FF7h
Vsync	Interrupt Vector # 2	0FF4h-0FF5h
Timer 1	Interrupt Vector # 3	0FF2h-0FF3h
PC4/PWRIN	Interrupt Vector # 4	0FF0h-0FF1h

Note: 1. This pin is associated with the NMI Interrupt Vector

Interrupt Vectors/Sources

The ST638x Core includes 5 different interrupt vectors in order to branch to 5 different interrupt routines. The interrupt vectors are located in the fixed (or static) page of the Program Space.

The interrupt vector associated with the non-maskable interrupt source is named interrupt vector #0. It is located at the (FFCh,FFDh) addresses in the Program Space. This vector is associated with the PC6/IRIN pin.

The interrupt vectors located at addresses (FF6h,FF7h), (FF4h,FF5h), (FF2h,FF3h), (FF0h,FF1h) are named interrupt vectors #1, #2, #3 and #4 respectively. These vectors are associated with TIMER 2 (#1), VSYNC (#2), TIMER 1 (#3) and PC4(PWRIN) (#4).

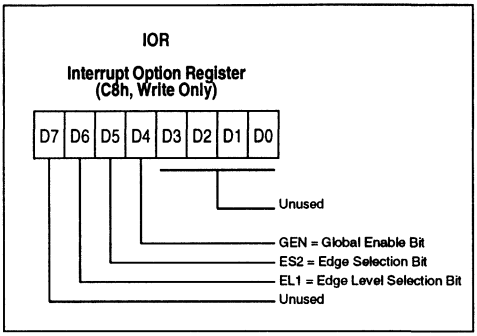
Interrupt Priority

The non-maskable interrupt request has the highest priority and can interrupt any other interrupt routines at any time, nevertheless the other interrupts cannot interrupt each other. If more than one interrupt request is pending, they are processed by the ST638x Core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is hardware fixed.

Interrupt Option Register

The Interrupt Option Register (IOR register, location C8h) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register can be addressed in the Data Space as RAM location at the C8h address, nevertheless it is write-only register that can not be accessed with single-bit operations. The operating modes of the external interrupt inputs associated to interrupt vectors #1 and #2 are selected through bits 5 and 6 of the IOR register.

Figure 19. Interrupt Option Register



D7. Not used.

EL1. This is the Edge/Level selection bit of interrupt #1. When set to one, the interrupt is generated on low level of the related signal; when cleared to zero, the interrupt is generated on falling edge. The bit is cleared to zero after reset.

ES2. This is the edge selection bit on interrupt #2. This bit is used on the ST638x devices with on-chip OSD generator for VSYNC detection.

GEN. This is the global enable bit. When set to one all interrupts are globally enabled; when this bit is cleared to zero all interrupts are disabled (excluding NMI).

D3 - D0. These bits are not used.

INTERRUPT (Continued)

Interrupt Procedure

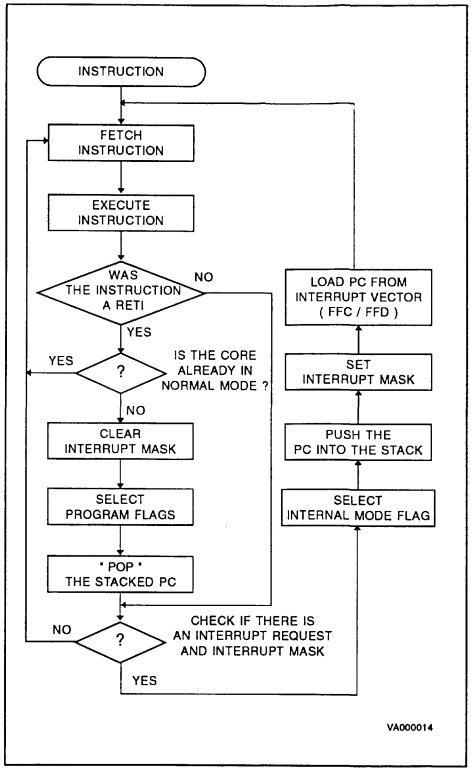
The interrupt procedure is very similar to a call procedure; the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event the user does not know about the context and the time at which it occurred. As a result the user should save all the data space registers which will be used inside the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes which are automatically switched and so these do not need to be saved.

The following list summarizes the interrupt procedure (refer also to Figure 20. Interrupt Processing Flow Chart):

- Interrupt detection
- The flags C and Z of the main routine are exchanged with the flags C and Z of the interrupt routine (resp. the NMI flags)
- The value of the PC is stored in the first level of the stack - The normal interrupt lines are inhibited (NMI still active)
- The edge flip-flop is reset
- The related interrupt vector is loaded in the PC.
- User selected registers are saved inside the interrupt service routine (normally on a software stack)
- The source of the interrupt is found by polling (if more than one source is associated to the same vector)
- Interrupt servicing
- Return from interrupt (RETI)
- Automatically the ST638x core switches back to the normal flags (resp the interrupt flags) and pops the previous PC value from the stack

The interrupt routine begins usually by the identification of the device that has generated the interrupt request. The user should save the registers which are used inside the interrupt routine (that holds relevant data) into a software stack.
After the RETI instruction execution, the Core carries out the previous actions and the main routine can continue.

Figure 20. Interrupt Processing Flow-Chart



ST638x Interrupt Details

IR Interrupt (#0). The IRIN/PC6 Interrupt is connected to the first interrupt #0 (NMI, 0FFCh). If the $\overline{\text{IRINT}}$ interrupt is disabled at the Latch circuitry, then it will be high. The #0 interrupt input detects a high to low level. Note that once #0 has been latched, then the only way to remove the latched #0 signal is to service the interrupt. #0 can interrupt the other interrupts. A simple latch is provided from the PC6(IRIN) pin in order to generate the $\overline{\text{IRINT}}$ signal. This latch can be triggered by either the positive or negative edge of IRIN signal. $\overline{\text{IRINT}}$ is inverted with respect to the latch. The latch can be read by software and reset by software.

INTERRUPT (Continued)

TIMER 2 Interrupt (#1). The TIMER 2 Interrupt is connected to the interrupt #1 (0FF6h). The TIMER 2 interrupt generates a low level (which is latched in the timer). Only the low level selection for #1 can be used. Bit 6 of the interrupt option register C8h has to be set.

VSYNC Interrupt (#2). The VSYNC Interrupt is connected to the interrupt #2. When disabled the VSYNC INT signal is low. The VSYNC INT signal is inverted with respect to the signal applied to the VSYNC pin. Bit 5 of the interrupt option register C8h is used to select the negative edge (ES2=0) or the positive edge (ES2=1); the edge will depend on the application. Note that once an edge has been latched, then the only way to remove the latched signal is to service the interrupt. Care must be taken not to generate spurious interrupts. This interrupt may be used for synchronize to the VSYNC signal in order to change characters in the OSD only when the screen is on vertical blanking (if desired). This method may also be used to blink characters.

TIMER 1 Interrupt (#3). The TIMER 1 Interrupt is connected to the fourth interrupt #3 (0FF2h) which detects a low level (latched in the timer).

PWR Interrupt (#4). The PWR Interrupt is connected to the fifth interrupt #4 (0FF0h). If the PWRINT is disabled at the PWR circuitry, then it will be high. The #4 interrupt input detects a low level. A simple latch is provided from the PC4 (PWRIN)pin in order to generate the PWRINT signal. This latch can be triggered by either the positive or negative edge of the PWRIN signal. PWRINT is inverted with respect to the latch. The latch can be reset by software.

Notes Global disable does not reset edge sensitive interrupt flags. These edge sensitive interrupts become pending again when global disabling is released. Moreover, edge sensitive interrupts are stored in the related flags also when interrupts are globally disabled, unless each edge sensitive interrupt is also individually disabled before the interrupting event happens. Global disable is done by clearing the GEN bit of Interrupt option register, while any individual disable is done in the control register of the peripheral. The on-chip Timer peripherals have an interrupt request flag bit (TMZ), this bit is set to one when the device wants to generate an interrupt request and a mask bit (ETI) that must be set to one to allow the transfer of the flag bit to the Core.

RESET

The ST638x devices can be reset in two ways: by the external reset input (RESET) tied low and by the hardware activated digital watchdog peripheral.

RESET Input

The external active low reset pin is used to reset the ST638x devices and provide an orderly software startup procedure. The activation of the Reset pin may occur at any time in the RUN or WAIT mode. Even short pulses at the reset pin will be accepted since the reset signal is latched internally and is only cleared after 2048 clocks at the oscillator pin. The clocks from the oscillator pin to the reset circuitry are buffered by a schmitt trigger so that an oscillator in start-up conditions will not give spurious clocks. When the reset pin is held low, the external crystal oscillator is also disabled in order to reduce current consumption. The MCU is configured in the Reset mode as long as the signal of the RESET pin is low. The processing of the program is stopped and the standard Input/Output ports (port A, port B and port C) are in the input state. As soon as the level on the reset pin becomes high, the initialization sequence is executed. Refer to the MCU initialization sequence for additional information.

Watchdog Reset

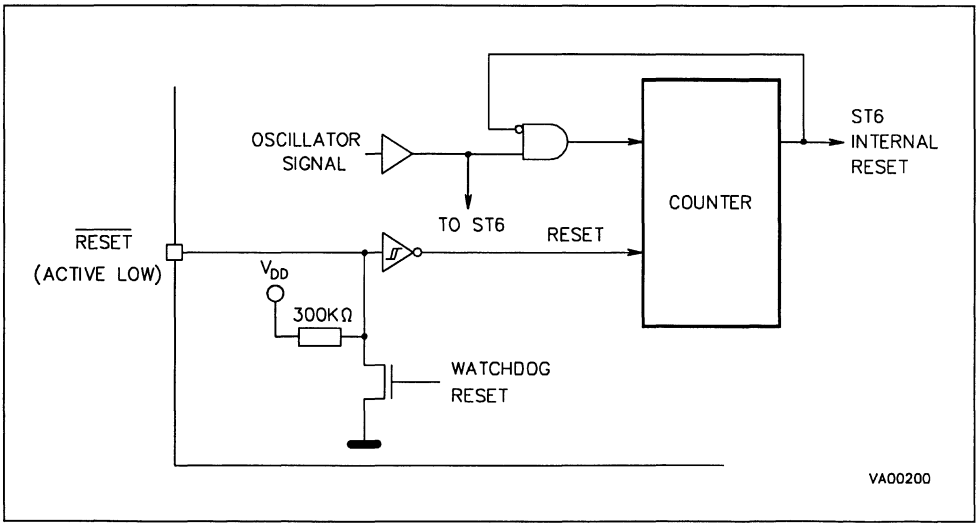
The ST638x devices are provided with an on-chip hardware activated digital watchdog function in order to provide a graceful recovery from a software upset. If the watchdog register is not refreshed and the end-of-count is reached, then the reset state will be latched into the MCU and an internal circuit pulls down the reset pin. This also resets the watchdog which subsequently turns off the pull-down and activates the pull-up device at the reset pin. This causes the positive transition at the reset pin. The MCU will then exit the reset state after 2048 clocks on the oscillator pin.

Application Notes

An external resistor between V_{DD} and the reset pin is not required because an internal pull-up device is provided. The user may prefer to add an external pull-up resistor.

An internal Power-on device does not guarantee that the MCU will exit the reset state when V_{DD} is above 4.5V and therefore the RESET pin should be externally controlled.

Figure 21. Internal Reset Circuit



RESET (Continued)

Figure 22. Reset & Interrupt Processing Flow-Chart

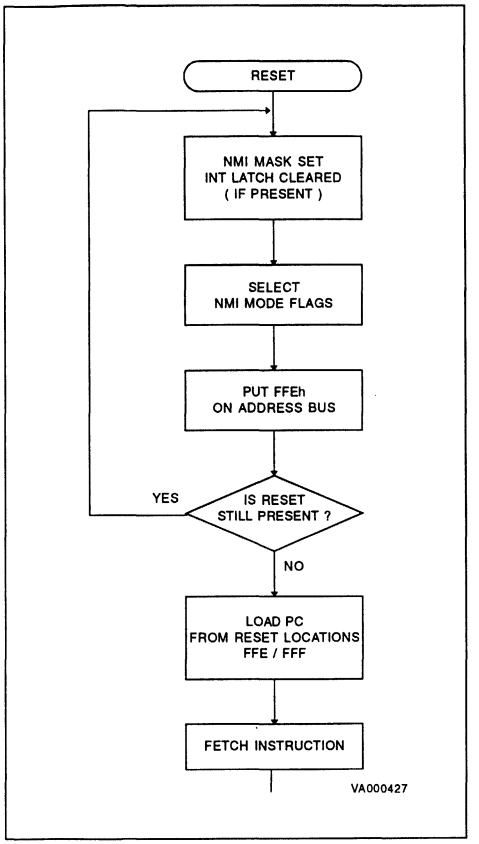
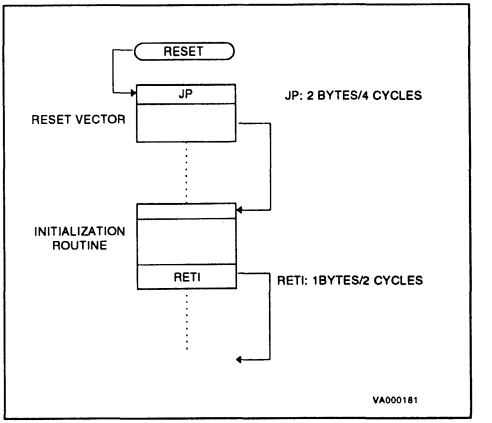


Figure 23. Restart Initialization Program Flow-Chart



MCU Initialization Sequence

When a reset occurs the stack is reset to program counter, the PC is loaded with the address of the reset vector (located in the program ROM at addresses FFEh & FFFh). A jump instruction to the beginning of the program has to be written into these locations. After a reset the interrupt mask is automatically activated so that the Core is in non-maskable interrupt mode to prevent false or ghost interrupts during the restart phase. Therefore the restart routine should be terminated by a RETI instruction to switch to normal mode and enable interrupts. If no pending interrupt is present at the end of the reset routine, the ST638x will continue with the instruction after the RETI; otherwise the pending interrupt will be serviced.

RESET Low Power Mode

When the reset pin is low, the quartz oscillator is Disabled allowing reduced current consumption. When the reset pin is raised the quartz oscillator is enabled and oscillations will start to build up.The internal reset circuitry will count 2048 clocks on the oscillator pin before allowing the MCU to go out of the reset state; the clocks are after a Schmitt trigger so that false or multiple counts are not possible.

WAIT & STOP MODES

The STOP and WAIT modes have been implemented in the ST638x Core in order to reduce the consumption of the device when the latter has no instruction to execute. These two modes are described in the following paragraphs. On ST638x as the hardware activated digital watchdog function is present the STOP instruction is de-activated and any attempt to execute it will cause the automatic execution of a WAIT instruction.

WAIT Mode

The configuration of the MCU in the WAIT mode occurs as soon as the WAIT instruction is executed. The microcontroller can also be considered as being in a "software frozen" state where the Core stops processing the instructions of the routine, the contents of the RAM locations and peripheral registers are saved as long as the power supply voltage is higher than the RAM retention voltage but where the peripherals are still working.
The WAIT mode is used when the user wants to reduce the consumption of the MCU when it is in idle, while not losing count of time or monitoring of external events. The oscillator is not stopped in order to provide clock signal to the peripherals. The timers counting may be enabled (writing the PSI bit in TSCR register) and the timer interrupt may be also enabled before entering the WAIT mode; this allows the WAIT mode to be left when timer interrupt occurs. If the exit from the WAIT mode is performed with a general RESET (either from the activation of the external pin or by watchdog reset) the MCU will enter a normal reset procedure as described in the RESET chapter. If an interrupt is generated during WAIT mode the MCU behaviour depends on the state of the ST638x Core before the initialization of the WAIT sequence, but also of the kind of the interrupt request that is generated. This case will be described in the following paragraphs. In any case, the ST638x Core does not generate any delay after the occurrence of the interrupt because the oscillator clock is still available.

STOP Mode

On ST638x the hardware watchdog is present and the STOP instruction has been de-activated. Any attempt to execute a STOP will cause the automatic execution of a WAIT instruction.

Exit from WAIT Mode

The following paragraphs describe the output procedure of the ST638x Core from WAIT mode when an interrupt occurs. It must be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) before the start of the WAIT sequence, but also of the type of the interrupt request that is generated. In all cases the GEN bit of IOR has to be set to 1 in order to restart from WAIT mode. Contrary to the operation of NMI in the RUN mode, the NMI is masked in WAIT mode if GEN=0.

Normal Mode. If the ST638x Core was in the main routine when the WAIT instruction has been executed, the ST638x Core outputs from the wait mode as soon as any interrupt occurs; the related interrupt routine is executed and at the end of the interrupt service routine the instruction that follows the WAIT instruction is executed if no other interrupts are pending.

Non-maskable Interrupt Mode. If the WAIT instruction has been executed during the execution of the non-maskable interrupt routine, the ST638x Core outputs from the wait mode as soon as any interrupt occurs: the instruction that follows the WAIT instruction is executed and the ST638x Core is still in the non-maskable interrupt mode even if another interrupt has been generated.

Normal Interrupt Mode. If the ST638x Core was in the interrupt mode before the initialization of the WAIT sequence, it outputs from the wait mode as soon as any interrupt occurs. Nevertheless, two cases have to be considered:

- If the interrupt is a normal interrupt, the interrupt routine in which the WAIT was entered will be completed with the execution of the instruction that follows the WAIT and the ST638x Core is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance to their priority.
- If the interrupt is a non-maskable interrupt, the non-maskable routine is processed at first. Then, the routine in which the WAIT was entered will be completed with the execution of the instruction that follows the WAIT and the ST638x Core is still in the normal interrupt mode.

Notes :
If all the interrupt sources are disabled, the restart of the MCU can only be done by a Reset activation. The Wait instruction is not executed if an enabled interrupt request is pending. In the ST638x the hardware activated digital watchdog function is present. As the watchdog is always activated the STOP instruction is de-activated and any attempt to execute the STOP instruction will cause an execution of a WAIT instruction.

ON-CHIP CLOCK OSCILLATOR

The internal oscillator circuit is designed to require a minimum of external components. A crystal quartz, a ceramic resonator, or an external signal (provided to the OSCin pin) may be used to generate a system clock with various stability/cost tradeoffs. The typical clock frequency is 8MHz. Please note that different frequencies will affect the operation of those peripherals (D/As, SPI) whose reference frequencies are derived from the system clock.

The different clock generator options connection methods are shown in Figures 24 and 25. One machine cycle takes 13 oscillator pulses; 12 clock pulses are needed to increment the PC while and additional 13th pulse is needed to stabilize the internal latches during memory addressing. This means that with a clock frequency of 8MHz the machine cycle is 1.625µs.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially RS), oscillator load capacitance (CL), IC parameters, ambient temperature, and supply voltage.It must be observed that the crystal or ceramic leads and circuit connections must be as short as possible. Typical values for CL1 and CL2 are in the range of 15pF to 22pF but these should be chosen based on the crystal manufacturers specification. Typical input capacitance for OSCin and OSCout pins is 5pF.

The oscillator output frequency is internally divided by 13 to produce the machine cycle and by 12 to produce the Timer and the Watchdog clock. A byte cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five byte cycles to be executed (See Table 7).

Table 7. Intructions Timing with 8MHz Clock

Instruction Type	Cycles	Execution Time
Branch if set/reset	5 Cycles	8.125µs
Branch & Subroutine Branch	4 Cycles	6.50µs
Bit Manipulation	4 Cycles	6.50µs
Load Instruction	4 Cycles	6.50µs
Arithmetic & Logic	4 Cycles	6.50µs
Conditional Branch	2 Cycles	3.25µs
Program Control	2 Cycles	3.25µs

Figure 24. Clock Generator Option (1)

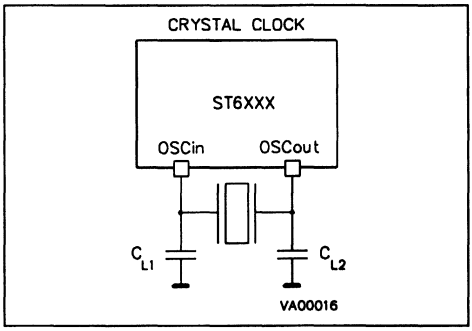


Figure 25. Clock Generator Option (2)

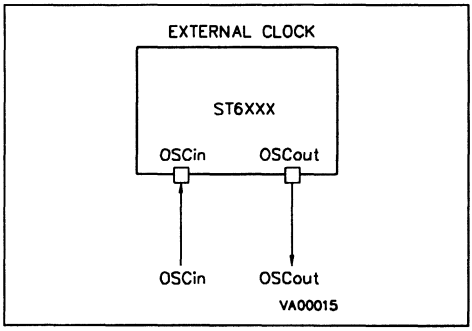
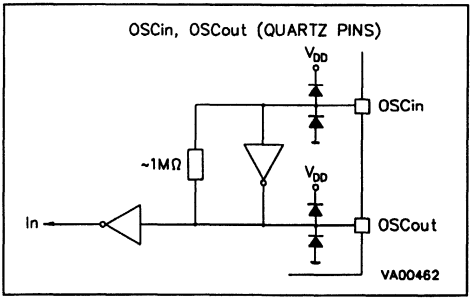


Figure 26. OSCIN, OSCOUT Diagram



INPUT/OUTPUT PORTS

The ST638x microcontrollers use three standard I/O ports (A,B,C) with up to eight pins on each port; refer to the device pin configurations to see which pins are available.

Each line can be individually programmed either in the input mode or the output mode as follows by software.

- Output
- Input with on-chip pull-up resistor (selected by software)
- Input without on-chip pull-up resistor (selected by software)

Note: pins with 12V open-drain capability do not have pull-up resistors.

In output mode the following hardware configurations are available:

- Open-drain output 12V (PA4-PA7, PC4-PC7)
- Open-drain output 5V (PC0-PC3)
- Push-pull output (PA0-PA3, PB0-PB6)

The lines are organized in three ports (port A,B,C). The ports occupy 6 registers in the data space. Each bit of these registers is associated with a particular line (for instance, the bits 0 of the Port A Data and Direction registers are associated with the PA0 line of Port A).

There are three Data registers (DRA, DRB, DRC), that are used to read the voltage level values of the lines programmed in the input mode, or to write the logic value of the signal to be output on the lines configured in the output mode. The port Data Registers can be read to get the effective logic levels of the pins, but they can be also written by the user software, in conjunction with the related Data Direction Register, to select the different input mode options. Single-bit operations on I/O registers (bit set/reset instructions) are possible but care is necessary because reading in input mode is made from I/O pins and therefore might be influenced by the external load, while writing will directly affect the Port data register causing an undesired changes of the input configuration. The three Data Direction registers (DDRA, DDRB, DDRC) allow the selection of the direction of each pin (input or output).

All the I/O registers can be read or written as any other RAM location of the data space, so no extra RAM cell is needed for port data storing and manipulation. During the initialization of the MCU, all the I/O registers are cleared and the input mode with pull-up is selected on all the pins thus avoiding pin conflicts (with the exception of PC2 that is set in output mode and is set high ie. high impedance).

Details of I/O Ports

When programmed as an input a pull-up resistor (if available) can be switched active under program control. When programmed as an output the I/O port will operate either in the push-pull mode or the open-drain mode according to the hardware fixed configuration as specified below.

Port A. PA0-PA3 are available as push-pull when outputs. PA4-PA7 are available as open-drain (no push-pull programmability) capable of withstanding 12V (no resistive pull-up in input mode). PA6-PA7 has been specially designed for higher driving capability and are able to sink 25mA with a maximum V_{OL} of 1V.

Port B. All lines are configured as push-pull when outputs.

Port C. PC0-PC3 are available as open-drain capable of withstanding a maximum V_{DD}+0.3V. PC4-PC7 are available as open-drain capable of withstanding 12V (no resistive pull-up in input mode). Some lines are also used as I/O buffers for signals coming from the on-chip SPI.

In this case the final signal on the output pin is equivalent to a wired AND with the programmed data output.

If the user needs to use the serial peripheral, the I/O line should be set in output mode while the open-drain configuration is hardware fixed; the corresponding data bit must set to one. If the latched interrupt functions are used (IRIN, PWRIN) then the corresponding pins should be set to input mode.

On ST638x the I/O pins with double or special functions are:

- PC0/SCL (connected to the SPI clock signal)
- PC1/SDA (connected to the SPI data signal)
- PC3/SEN (connected to the SPI enable signal)
- PC4/PWRIN (connected to the PWRIN interrupt latch)
- PC6/IRIN (connected to the IRIN interrupt latch)

All the Port A,B and C I/O lines have Schmitt-trigger input configuration with a typical hysteresis of 1V.

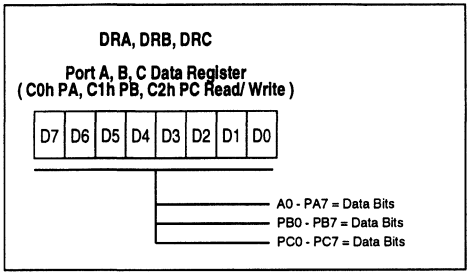
INPUT/OUTPUT PORTS (Continued)

Table 8. I/O Port Options Selection

DDR	DR	Mode	Option
0	0	Input	With on-chip pull-up resistor
0	1	Input	Without on-chip pull-up resistor
1	X	Output	Open-drain or Push-Pull

Note: X: Means don't care.

Figure 27. Port A, B, C Data Register



PA7-PA0. These are the I/O port A data bits. Reset at power-on.

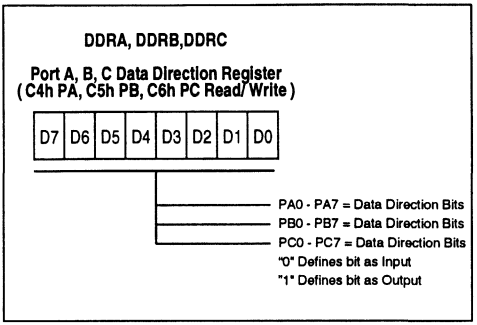
PB7-PB0. These are the I/O port B data bits. Reset at power-on.

PC7-PC0. Set to 04h at power-on. Bit 2 (PC2 pin) is set to one (open drain therefore high impedence).

I/O Pin Programming

Each pin can be individually programmed as input or output with different input and output configurations. This is achieved by writing to the relevant bit in the data (DR) and data direction register (DDR). Table 8 shows all the port configurations that can be selected by the user software.

Figure 28. Port A, B, C Data Register



PA7-PA0. These are the I/O port A data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Reset at power-on.

PB7-PB0. These are the I/O port B data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Reset at power-on.

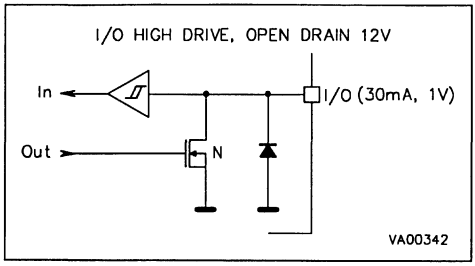
PC7-PC0. These are the I/O port C data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Set to 04h at power-on. Bit 2 (PC2 pin) is set to one (output mode selected).

INPUT/OUTPUT PORTS (Continued)

Input/Output Configurations

The following schematics show the I/O lines hardware configuration for the different options. Figure 29 shows the I/O configuration for an I/O pin with open-drain 12V capability (standard drive and high drive). Figure 30 shows the I/O configuration for an I/O pin with push-pull and with open drain 5V capability.

Figure 29. I/O Configuration Diagram (Open Drain 12V)

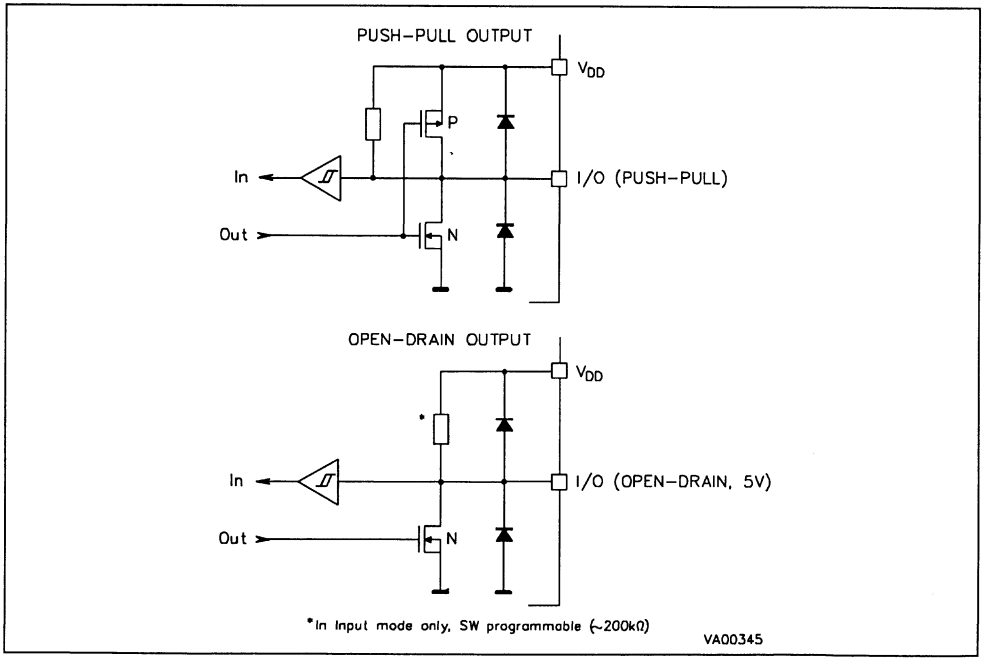


Notes :

The WAIT instruction allows the ST638x to be used in situations where low power consumption is needed. This can only be achieved however if the I/O pins either are programmed as inputs with well defined logic levels or have no power consuming resistive loads in output mode. As the same die is used for the different ST638x versions the unavailable I/O lines of ST638x should be programmed in output mode.

Single-bit operations on I/O registers are possible but **care is necessary** because reading in input mode is made from I/O pins while writing will directly affect the Port data register causing an undesired changes of the input configuration.

Figure 30. I/O Configuration Diagram (Open Drain 5V, Push-pull)



TIMERS

The ST638x devices offer two on-chip Timer peripherals consisting of an 8-bit counter with a 7-bit programmable prescaler, thus giving a maximum count of 2^{15}, and a control logic that allows configuring the peripheral operating mode. Figure 30 shows the timer block diagram. The content of the 8-bit counters can be read/written in the Timer/Counter registers TCR that can be addressed in the data space as RAM location at addresses D3h (Timer 1) and DBh (Timer 2). The state of the 7-bit prescaler can be read in the PSC register at addresses D2h (Timer 1) and DAh (Timer 2). The control logic is managed by TSCR registers at D4h (Timer 1) and DCh (Timer 2) addresses as described in the following paragraphs.

The following description applies to both Timer 1 and Timer 2. The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (timer zero) bit in the TSCR is set to one. If the ETI (enable timer interrupt) bit in the TSCR is also set to one an interrupt request, associated to interrupt vector #3 (for Timer 1) and #1 for Timer 2, is generated. The interrupt of the timer can be used to exit the MCU from the WAIT mode.

The prescaler decrements on rising edge. The prescaler input is the oscillator frequency divided by 12.

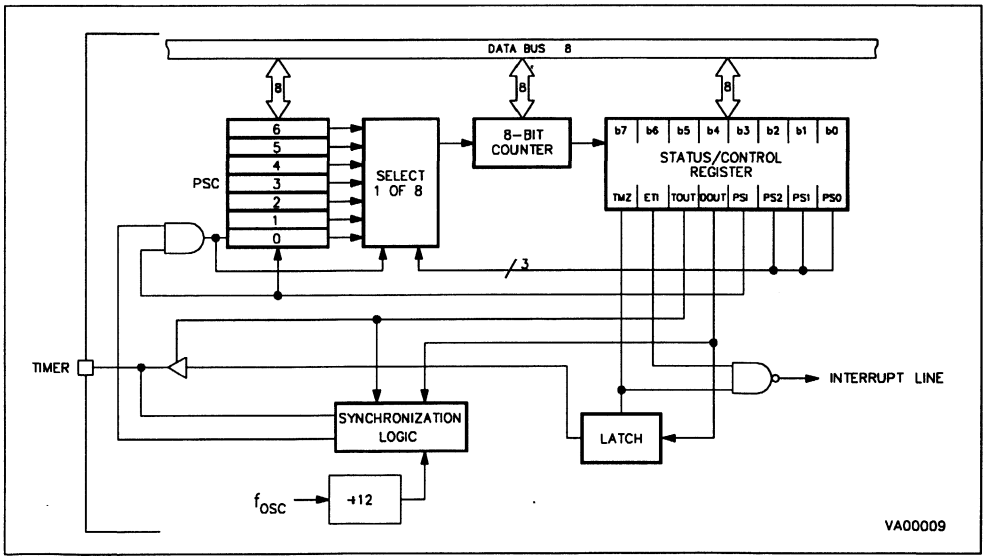
Depending on the division factor programmed by PS2/PS1/PS0 (see table 9) bits in the TSCR, the clock input of the timer/counter register is multiplexed to different sources.

On division factor 1, the clock input of the prescaler is also that of timer/counter; on factor 2, bit 0 of prescaler register is connected to the clock input of TCR.

This bit changes its state with the half frequency of prescaler clock input. On factor 4, bit 1 of PSC is connected to clock input of TCR, and so on. On division factor 128, the MSB bit 6 of PSC is connected to clock input of TCR. The prescaler initialize bit (PSI) in the TSCR register must be set to one to allow the prescaler (and hence the counter) to start. If it is cleared to zero then all of the prescaler bits are set to one and the counter is inhibited from counting.

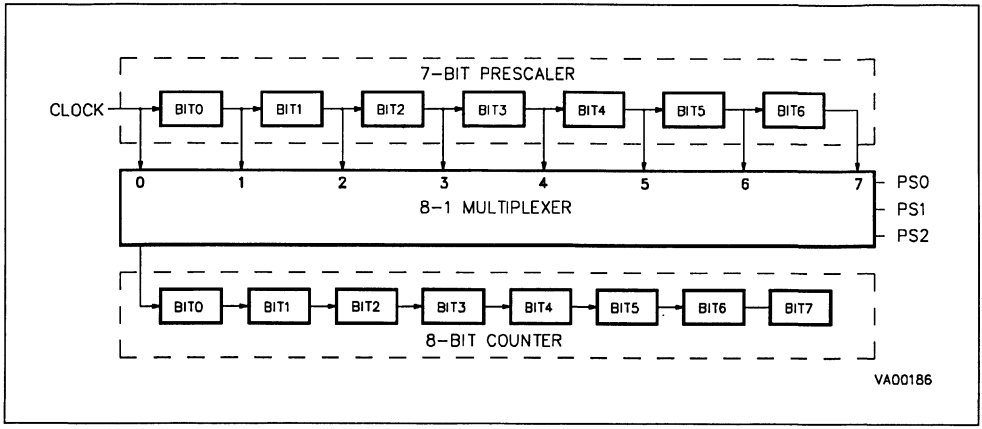
The prescaler can be given any value between 0 and 7Fh by writing to the related register address, if bit PSI in the TSCR register is set to one. The tap of the prescaler is selected using the PS2/PS1/PS0 bits in the control register. Figure 31 shows the timer working principle.

Figure 30. Timer Peripheral Block Diagram



TIMERS (Continued)

Figure 31. Timer Working Principle



Timer Operating Modes

As on ST638x devices the external TIMER pin is not available the only allowed operating mode is the output mode that have to be selected by setting to 1 bit 4 and by clearing to 0 bit 5 in the TSCR1 register. This procedure will enable both Timer 1 and Timer 2.

Output Mode (TSCR1 D4 = 1, TSCR1 D5 = 0). On this mode the timer prescaler is clocked by the prescaler clock input (OSC/12). The user can select the desired prescaler division ratio through the PS2/PS1/PS0 bits. When TCR count reaches 0, it sets the TMZ bit in the TSCR.

The TMZ bit can be tested under program control to perform timer functions whenever it goes high. Bit D4 and D5 on TSCR2 (Timer 2) register are not implemented.

Timer Interrupt

When the counter register decrements to zero and the software controlled ETI (enable timer interrupt) bit is set to one then an interrupt request associated to interrupt vector #3 (for Timer 1) and to interrupt vector #1 (for Timer 2) is generated. When the counter decrements to zero also the TMZ bit in the TSCR register is set to one.

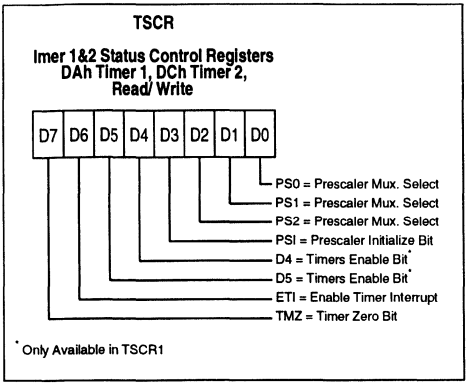
Notes :

TMZ is set when the counter reaches 00h ; however, it may be set by writing 00h in the TCR register or setting the bit 7 of the TSCR register. TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded to FFh while the 7-bit prescaler is loaded to 7Fh , and the TSCR register is cleared which means that timer is stopped (PSI=0) and timer interrupt disabled.

A write to the TCR register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR and the PSC registers can be read accurately at any time.

TIMERS (Continued)

Figure 32. Timer Status Control Registers



TMZ. Low-to-high transition indicates that the timer count register has decrement to zero. This bit must be cleared by user software before to start with a new count.

ETI. This bit, when set, enables the timer interrupt (vector #3 for Timer 1, vector #1 for Timer 2) request. If ETI=0 the timer interrupt is disabled. If ETI= 1 and TMZ= 1 an interrupt request is generated.

D5. This is the timers enable bit D5. It must be cleared to 0 together with a set to 1 of bit D4 to enable both Timer 1 and Timer 2 functions. It is not implemented on TSCR2 register.

D4. This is the timers enable bit D4. This bit must be set to 1 together with a clear to 0 of bit D5 to enable both Timer 1 and Timer 2 functions. It is not implemented on TSCR2 register.

D5	D4	Timers
0	0	Disabled
0	1	Enabled
1	X	Reserved

PS1. Used to initialize the prescaler and inhibit its counting while PSI = 0 the prescaler is set to 7Fh and the counter is inhibited. When PSI = 1 the prescaler is enabled to count downwards. As long as PSI= 0 both counter and prescaler are not running.

PS2-PS0. These bits select the division ratio of the prescaler register. (see table 9)

The TSCR1 and TSCR2 registers are cleared on reset. The correct D4-D5 combination must be written in TSCR1 by user's software to enable the operation of Timer 1 and Timer 2.

Table 9. Prescaler Division Factors

PS2	PS1	PS0	Divided By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Figure 33. Timer Counter Registers

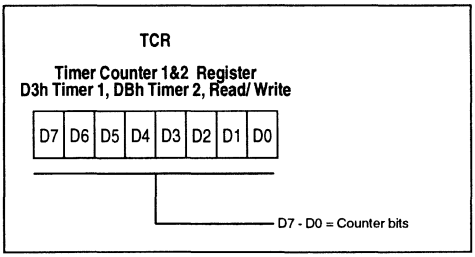
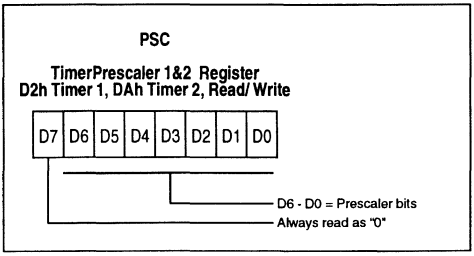


Figure 34. Timer Counter Registers



HARDWARE ACTIVATED DIGITAL WATCHDOG FUNCTION

The hardware activated digital watchdog function consists of a down counter that is automatically initialized after reset so that this function does not need to be activated by the user program. As the watchdog function is always activated this down counter can not be used as a timer. The watchdog is using one data space register (HWDR location D8h). The watchdog register is set to FEh on reset and immediately starts to count down, requiring no software start. Similarly the hardware activated watchdog can not be stopped or delayed by software.

The watchdog time can be programmed using the 6 MSbits in the watchdog register, this gives the possibility to generate a reset in a time between 3072 to 196608 oscillator cycles in 64 possible steps. (With a clock frequency of 8MHz this means from 384µs to 24:576ms). The reset is prevented if the register is reloaded with the desired value before bits 2-7 decrement from all zeros to all ones.

The presence of the hardware watchdog deactivates the STOP instruction and a WAIT instruction is automatically executed instead of a STOP. Bit 1 of the watchdog register (set to one at reset) can be used to generate a software reset if cleared to zero). Figure 35 shows the watchdog block diagram while Figure 36 shows its working principle.

Figure 36. Hardware Activated Watchdog Working Principle

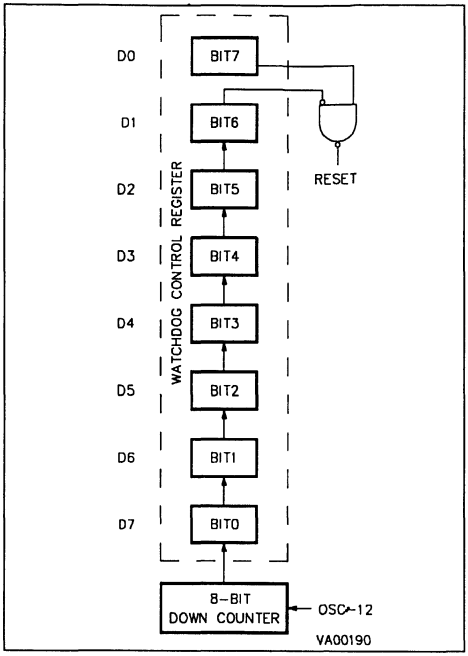
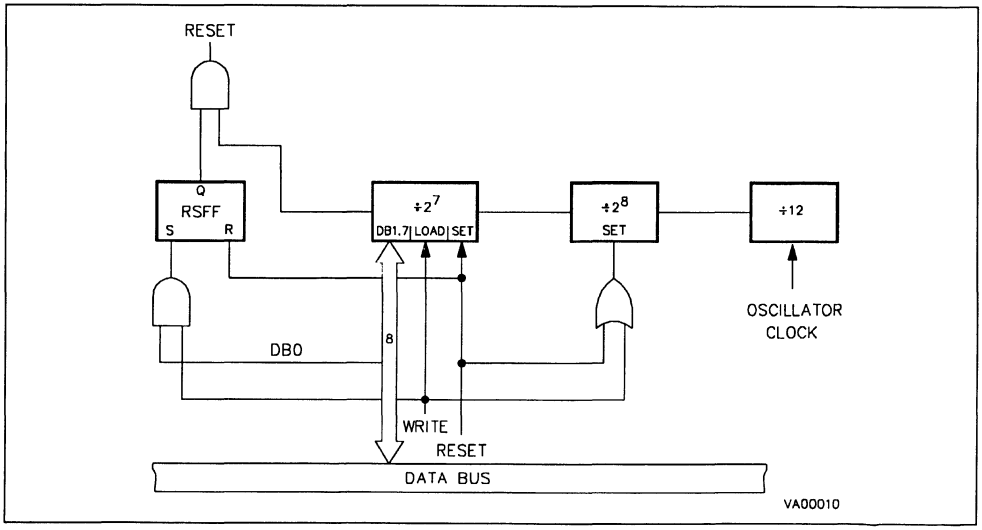
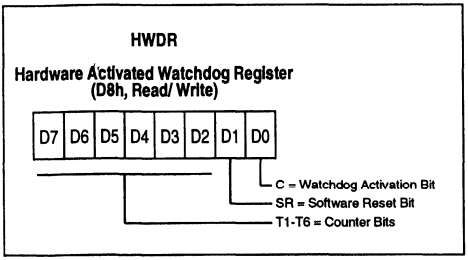


Figure 35. Hardware Activated Watchdog Block Diagram



HARDWARE ACTIVATED DIGITAL WATCHDOG FUNCTION (Continued)

Figure 37. Watchdog Register



T1-T6. These are the watchdog counter bits. It should be noted that D7 (T1) is the LSB of the counter and D2 (T6) is the MSB of the counter, these bits are in the opposite order to normal.

SR. This bit is set to one during the reset phase and will generate a software reset if cleared to zero.

C. This is the watchdog activation bit that is hardware set. The watchdog function is always activated independently of changes of value of this bit.

The register reset value is FEh (Bit 1-7 set to one, Bit 0 cleared).

SERIAL PERIPHERAL INTERFACE

The ST638x Serial Peripheral Interface (SPI) has been designed to be cost effective and flexible in interfacing the various peripherals in TV applications.

It maintains the software flexibility but adds hardware configurations suitable to drive devices which require a fast exchange of data. The three pins dedicated for serial data transfer (single master only) can operate in the following ways:

- as standard I/O lines (software configuration)
- as S-BUS or as I^2CBUS (two pins)
- as standard (shift register) SPI

When using the hardware SPI, a fixed clock rate of 62.5kHz is provided.

It has to be noted that the first bit that is output on the data line by the 8-bit shift register is the MSB.

SPI Data/Control Registers

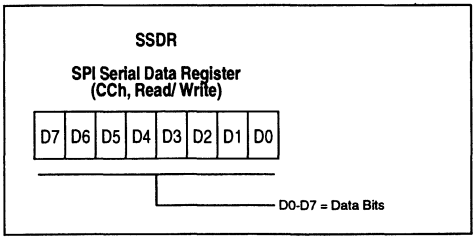
For I/O details on SCL (Serial Clock), SDA (Serial Data) and SEN (Serial Enable) please refer to I/O Ports description with reference to the following registers:

Port C data register, Address C2h (Read/Write).

- BIT D0 "SCL"
- BIT D1 "SDA"
- BIT D3 "SEN"

Port C data direction register, Address C6h (Read/Write).

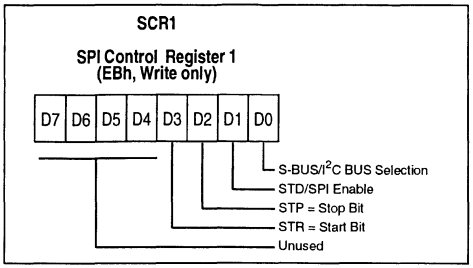
Figure 38. SPI Serial Data Register



D7-D0. These are the SPI data bits. They can be neither read nor written when SPI is operating (BUSY bit set). They are undefined after reset.

SERIAL PERIPHERAL INTERFACE (Continued)

Figure 39. SPI Control Register 1



D7-D4. These bits are not used.

STR. This is Start bit for I²CBUS/S-BUS. This bit is meaningless when STD/SPI enable bit is cleared to zero. If this bit is set to one STD/SPI bit is also set to "1" and SPI Start generation, before beginning of transmission, is enabled. Set to zero after reset.

STP. This is Stop bit for I²CBUS/S-BUS. This bit is meaningless when STD/SPI enable bit is cleared to zero. If this bit is set to one STD/SPI bit is also set to "1" and SPI Stop condition generation is enabled. STP bit must be reset when standard protocol is used (this is also the default reset conditions). Set to zero after reset.

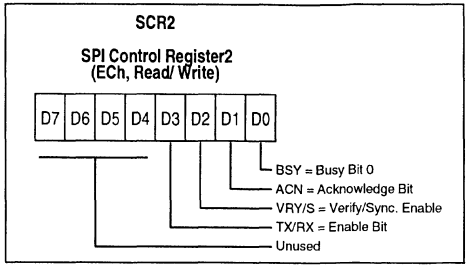
STD, SPI Enable. This bit, in conjunction with S-BUS/I²CBUS bit, allows the SPI disable and will select between I²CBUS/S-BUS and Standard shift register protocols. If this bit is set to one, it selects both I²CBUS and S-BUS protocols; final selection between them is made by S-BUS/I²CBUS bit. If this bit is cleared to zero when S-BUS/I²CBUS is set to "1" the Standard shift register protocol is selected. If this bit is cleared to "0" when S-BUS/I²CBUS is cleared to 0 the SPI is disabled. Set to zero after reset.

S-BUS/I²CBUS Selection. This bit, in conjunction with STD/SPI bit, allows the SPI disable and will select between I²CBUS and S-BUS protocols. If this bit is cleared to "0" when STD bit is also "0", the SPI interface is disabled. If this bit is cleared to zero when STD bit is set to "1", the I²CBUS protocol will be selected. If this bit is set to "1" when STD bit is set to "1", the S-BUS protocol will be selected. Cleared to zero after reset.

Table 10. SPI Modes Selection

D1 STD/SP	D0 S-BUS/I²C BUS	SPI Function
0	0	Disabled
0	1	STD Shift Reg.
1	0	I²C BUS
1	1	S-BUS

Figure 40. SPI Control Register2



D7-D4. These bits are not used.

TX/RX. *Write Only.* When this bit is set, current byte operation is a transmission. When it is reset, current operation is a reception. Set to zero after reset.

VRY/S. *Read Only/Write Only.* This bit has two different functions in relation to read or write operation. *Reading Operation:* when STD and/or TRX bits is cleared to 0, this bit is meaningless. When bits STD and TX are set to 1, this bit is set each time BSY bit is set. This bit is reset during byte operation if real data on SDA line are different from the output from the shift register. Set to zero after reset. *Writing Operation* : it enables (if set to one) or disables (if cleared to zero) the interrupt coming from VSYNC pin. Undefined after reset. Refer to OSD description for additional information.

ACN. *Read Only.* If STD bit (D1 of SCR1 register) is cleared to zero this bit is meaningless. When STD is set to one, this bit is set to one if no Acknowledge has been received. In this case it is automatically reset when BSY is set again. Set to zero after reset.

BSY. *Read/Set Only.* This is the busy bit. When a one is loaded into this bit the SPI interface start the transmission of the data byte loaded into SSDR data register or receiving and building the receive data into the SSDR data register. This is done in accordance with the protocol, direction and start/stop condition(s). This bit is automatically cleared at the end of the current byte operation. Cleared to zero after reset.

Note :
The SPI shift register is also the data transmission register and the data received register; this feature is made possible by using the serial structure of the ST638x and thus reducing size and complexity.

SERIAL PERIPHERAL INTERFACE (Continued)

During transmission or reception of data, all access to serial data register is therefore disabled. The reception or transmission of data is started by setting the BUSY bit to "1"; this will be automatically reset at the end of the operation. After reset, the busy bit is cleared to "0", and the hardware SPI disabled by clearing bit 0 and bit 1 of SPI control register 1 to "0". The outputs from the hardware SPI are "ANDed" to the standard I/O software controlled outputs. If the hardware SPI is in operation the Port C pins related to the SPI should be configured as outputs using the Data Direction Register and should be set high. When the SPI is configured as the S-BUS, the three pins PC0, PC1 and PC3 become the pins SCL, SDA and SEN respectively. When configured as the I^2CBUS the pins PC0 and PC1 are configured as the pins SCL and SDA; PC3 is not driven and can be used as a general purpose I/O pin. In the case of the STD SPI the pins PC0 and PC1 become the signals CLOCK and DATA, PC3 is not driven and can be used as general purpose I/O pin. The VERIFY bit is available when the SPI is configured as either S-BUS or I^2CBUS. At the start of a byte transmission, the verify bit is set to one. If at any time during the transmission of the following eight bits, the data on the SDA line does not match the data forced by the SPI (while SCL is high), then the VERIFY bit is reset. The verify is available only during transmission for the S-BUS and I^2CBUS; for other protocol it is not defined. The SDA and SCL signal entering the SPI are buffered in order to remove any minor glitches. When STD bit is set to one (S-BUS or I^2CBUS selected), and TRX bit is reset (receiving data), and STOP bit is set (last byte of current communication), the SPI interface does not generate the Acknowledge, according to S-BUS/I^2CBUS specifications. PCO-SCL, PC1-SDA and PC3-SEN lines are standard drive I/O port pins with open-drain output configuration (maximum voltage that can be applied to these pins is V_{DD}+ 0.3V).

S-BUS/I^2CBUS Protocol Information

The S-BUS is a three-wire bidirectional data-bus with functional features similar to the I^2CBUS. In fact the S-BUS includes decoding of Start/Stop conditions and the arbitration procedure in case of multimaster system configuration (the ST638x SPI allows a single-master only operation). The SDA line, in the I^2CBUS represents the AND combination of SDA and SEN lines in the S-BUS. If the SDA and the SEN lines are short-circuit connected, they appear as the SDA line of the I^2CBUS. The Start/Stop conditions are detected (by the external peripherals suited to work with S-BUS/I^2CBUS) in the following way:

- On S-BUS by a transition of the SEN line (1 to 0 Start, 0 to 1 Stop) while the SCL line is at high level.
- On I^2CBUS by a transition of the SDA line (10 Start, 01Stop) while the SCL line is at high level.

Start and Stop condition are always generated by the master (ST638x SPI can only work as single master). The bus is busy after the start condition and can be considered again free only when a certain time delay is left after the stop condition. In the S-BUS configuration the SDA line is only allowed to change during the time SCL line is low. After the start information the SEN line returns to high level and remains unchanged for all the data transmission time. When the transmission is completed the SDA line is set to high level and, at the same time, the SEN line returns to the low level in order to supply the stop information with a low to high transition, while the SCL line is at high level. On the S-BUS, as on the I^2CBUS, each eight bit information (byte) is followed by one acknowledged bit which is a high level put on the SDA line by the transmitter. A peripheral that acknowledges has to pull down the SDA line during the acknowledge clock pulse. An addressed receiver has to generate an acknowledge after the reception of each byte; otherwise the SDA line remains at the high level during the ninth clock pulse time. In this case the master transmitter can generate the Stop condition, via the SEN (or SDA in I^2CBUS) line, in order to abort the transfer.

SERIAL PERIPHERAL INTERFACE (Continued)

Start/Stop Acknowledge. The timing specs of the S-BUS protocol require that data on the SDA (only on this line for I^2CBUS) and SEN lines be stable during the "high" time of SCL. Two exceptions to this rule are foreseen and they are used to signal the start and stop condition of data transfer.

- On S-BUS by a transition of the SEN line (10 Start, 01 Stop) while the SCL line is at high level.
- On I^2CBUS by a transition of the SDA line (10 Start, 01 Stop) while the SCL line is at high level.

Data are transmitted in 8-bit groups; after each group, a ninth bit is interposed, with the purpose of acknowledging the transmitting sequence (the transmit device place a "1" on the bus, the acknowledging receiver a "0").

Interface Protocol. This paragraph deals with the description of data protocol structure. The interface protocol includes:

- A start condition

- A "slave chip address" byte, transmitted by the master, containing two different information:

a. the code identifying the device the master wants to address (this information is present in the first seven bits)

b. the direction of transmission on the bus (this information is given in the 8th bit of the byte); "0" means "Write", that is from the master to the slave, while "1" means "Read". The addressed slave must always acknowledge.

The sequence from, now on, is different according to the value of R/$\overline{W}$ bit.

1. R/$\overline{W}$ = "0" ($\overline{\text{Write}}$)

In all the following bytes the master acts as transmitter; the sequence follows with:

a. an optional data byte to address (if needed) the slave location to be written (it can be a word address in a memory or a register address, etc.).

b. a "data" byte which will be written at the address given in the previous byte.

c. further data bytes.

d. a STOP condition

A data transfer is always terminated by a stop condition generated from the master. The ST638x peripheral must finish with a stop condition before another start is given. Figure 44 shows an example of write operation.

2. R/$\overline{W}$ = "1" (Read)

In this case the slave acts as transmitter and, therefore, the transmission direction is changed. In read mode two different conditions can be considered:

a. The master reads slave immediately after first byte. In this case after the slave address sent from the master with read condition enabled the master transmitter becomes master receiver and the slave receiver becomes slave transmitter.

b. The master reads a specified register or location of the slave. In this case the first sent byte will contain the slave address with write condition enabled, then the second byte will specify the address of the register to be read. At this moment a new start is given together with the slave address in read mode and the procedure will proceed as described in previous point "a".

SERIAL PERIPHERAL INTERFACE (Continued)

Figure 41.Master Transmit to Slave Receiver (Write Mode)

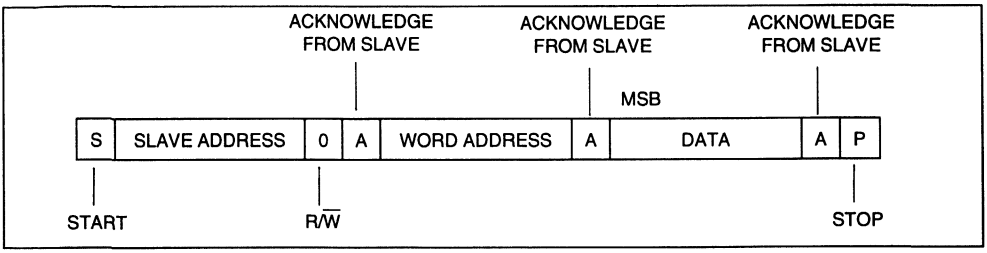


Figure 42.Master Reads Slave Immediately After First Byte (read Mode)

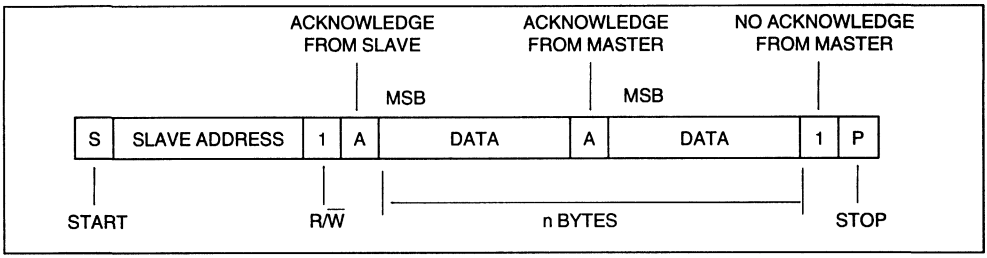
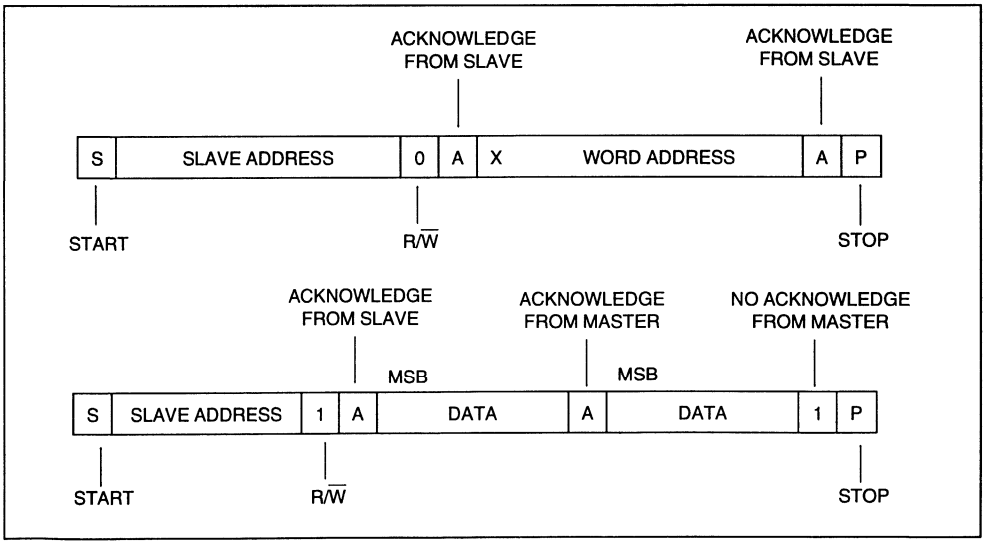


Figure 43.Master Reads After Setting Slave Register Address (Write Address, Read Data)

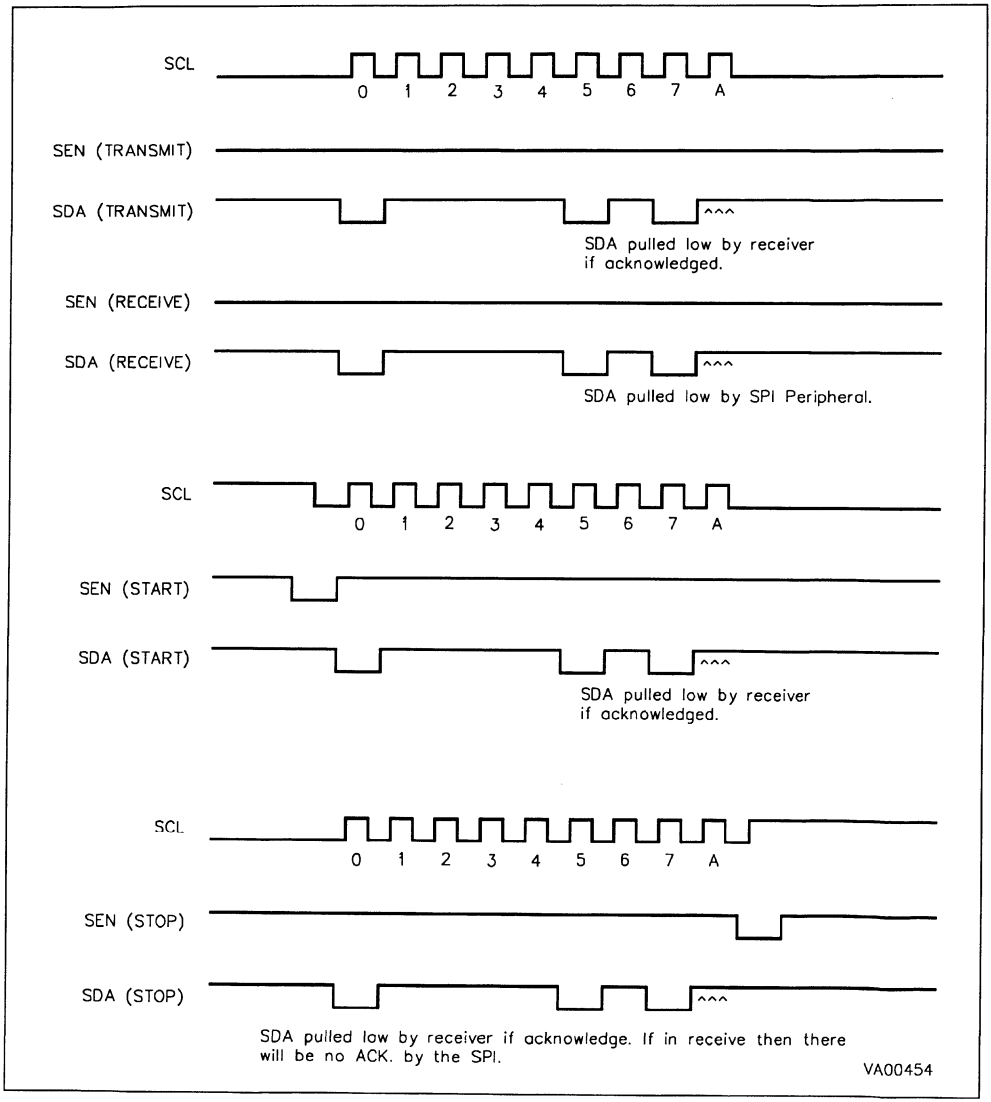


SERIAL PERIPHERAL INTERFACE (Continued)

S-BUS/I²CBUS Timing Diagrams

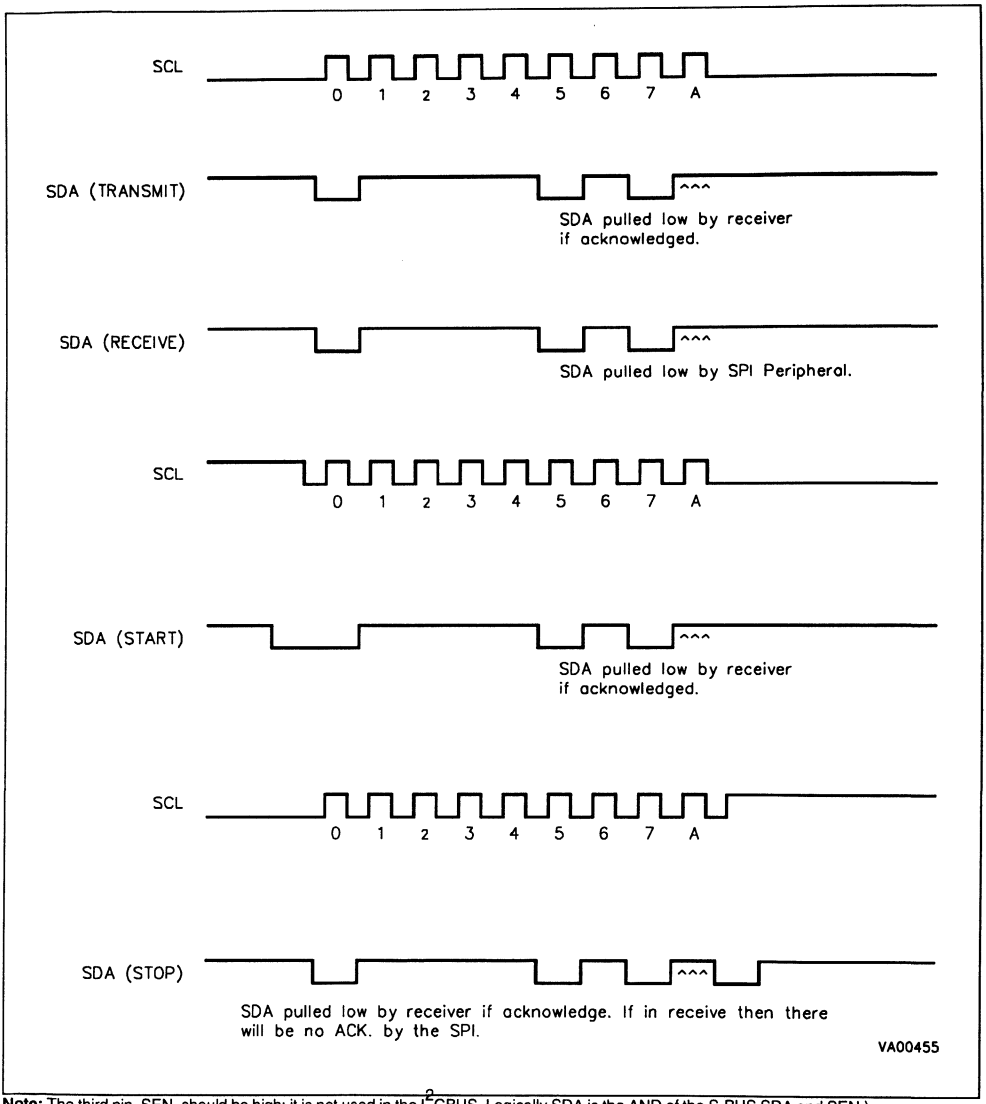
The clock of the S-BUS/I²CBUS of the ST638x SPI (single master only) has a fixed bus clock frequency of 62.5KHz. All the devices connected to the bus must be able to follow transfers with frequencies up to 62.5KHz, either by being able to transmit or receive at that speed or by applying the clock synchronization procedure which will force the master into a wait state and stretch low periods.

Figure 44. S-BUS Timing Diagram



SERIAL PERIPHERAL INTERFACE (Continued)

Figure 45. I²C BUS Timing Diagram



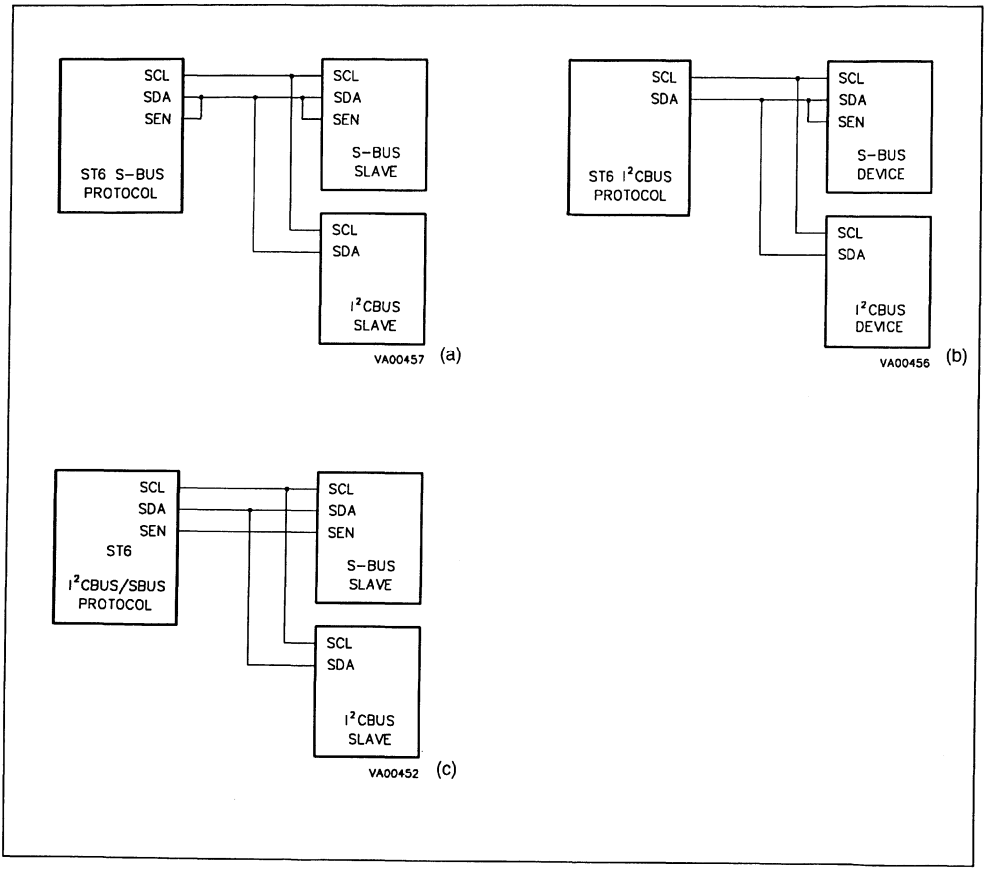
Note: The third pin, SEN, should be high; it is not used in the I²CBUS. Logically SDA is the AND of the S-BUS SDA and SEN.)

SERIAL PERIPHERAL INTERFACE (Continued)

Compatibility S-BUS/I²CBUS

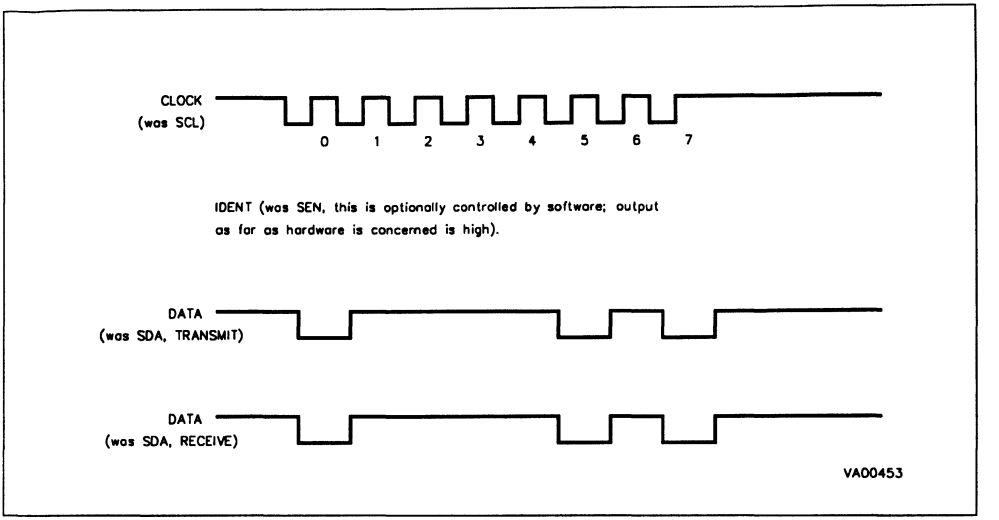
Using the S-BUS protocol it is possible to implement mixed system including S-BUS/I²CBUS bus peripherals. In order to have the compatibility with the I²CBUS peripherals, the devices including the S-BUS interface must have their SDA and SEN pins connected together as shown in the following Figure 46 (a and b). It is also possible to use mixed S-BUS/I²CBUS protocols as showed in figure 46 (c). S-BUS peripherals will only react to S-BUS protocol signals, while I²CBUS peripherals will only react to I²CBUS signals. Multimaster configuration is not possible with the ST63xx SPI (single master only).

Figure 46.S-BUS/I²C BUS Mixed Configurations



SERIAL PERIPHERAL INTERFACE (Continued)

Figure 47.Sofware Bus (Hardware Bus Disabled) Timing Diagram



STD SPI Protocol (Shift Register)

This protocol is similar to the I^2CBUS with the exception that there is no acknowledge pulse and there are no stop or start bits. The clock cannot be slowed down by the external peripherals.

In this case all three outputs should be high in order not to lock the software I/Os from functioning.

SPI Standard Bus Protocol: The standard bus protocol is selected by loading the SPI Control Register 1 (SCR1 Add. EBh). Bit 0 named I^2C must be set at one and bit 1 named STD mut be reset. When the standard bus protocol is selected bit 2 of the SCR1 is meaningless.

This bit named STOP bit is used only in I^2CBUS or SBUS. However take care thet THE STOP BIT MUST BE RESET WHEN THE STANDARD PROTOCOL IS USED. This bit is set to ZERO after RESET.

14-BIT VOLTAGE SYNTHESIS TUNING PERIPHERAL

The ST638x on-chip voltage synthesis tuning peripheral has been integrated to allow the generation of tuning reference voltage in low/mid end TV set applications. The peripheral is composed of a 14-bit counter that allows the conversion of the digital content in a tuning voltage, available at the VS output pin, by using Pulse Width Modification (PWM), and Bit Rate Multiplier (BRM) techniques. The 14-bit counter gives 16384 steps which allows a resolution of approximately 2mV over a tuning voltage of 32V; this corresponds to a tuning resolution of about 40KHz per step in the UHF band (the actual value will depend on the characteristics of the tuner).

The tuning word consists of a 14-bit word contained in the registers VSDATA1 (location 0EEh) and VSDATA2 (location 0EFh). Coarse tuning (PWM) is performed using the seven MSBits, while fine tuning (BRM) is performed using the data in the seven LSBits. With all zeros loaded the output is zero; as the tuning voltage increases from all zeros, the number of pulses in one period increas to 128 with all pulses being the same width. For values larger than 128, the PWM takes over and the number of pulses in one period remains constant at 128, but the width changes. At the other end of the scale, when almost all ones are loaded, the pulses will start to link together and the number of pulses will decrease. When all ones are loaded, the output will be almost 100% high but will have a low pulse (1/16384 of the high pulse).

Output Details

Inside the on-chip Voltage Synthesis are included the register latches, a reference counter, PWM and BRM control circuitry. In the ST638x the clock for the 14-bit reference counter is 2MHz derived from the 8MHz system clock. From the circuit point of view, the seven most significant bits control the coarse tuning, while the seven least significant bits control the fine tuning. From the application and software point of view, the 14 bits can be considered as one binary number.

As already mentioned the coarse tuning consists of a PWM signal with 128 steps ; we can consider the fine tuning to cover 128 coarse tuning cycles. The addition of pulses is described in the following Table.

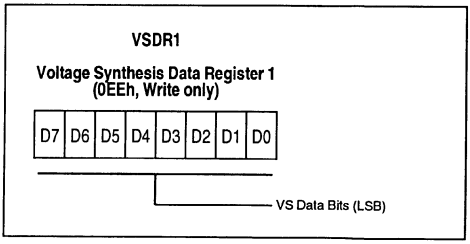
Table 11. Fine Tuning Pulse Addition

FIne Tuning (7 LSB)	N° of Pulses added at the following cycles (0...127)
0000001	64
0000010	32, 96
0000100	16, 48, 80, 112
0001000	8, 24, ...104, 120
0010000	4, 12, ...116, 124
0100000	2, 6, ...122, 126
1000000	1, 3, ...125, 127

The VS output pin has a standard drive push-pull output configuration.

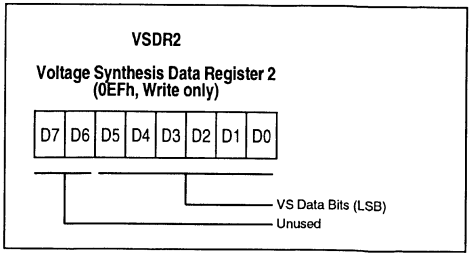
VS Tuning Cell Registers

Figure 48. Voltage Synthesis Data Register 1



D7-D0. These are the 8 least significant VS data bits. Bit 0 is the LSB. This register is undefined on reset.

Figure 49.Voltage Synthesis Data Register 2



D7-D6. These bits are not used.

D5-D0. These are the 6 most significant VS data bits. Bit 5 is the MSB. This register is undefined on reset.

6-BIT PWM D/A CONVERTERS

The D/A macrocell contains up to six PWM D/A outputs (31.25kHz repetition, DA0-DA5) with six bit resolution.

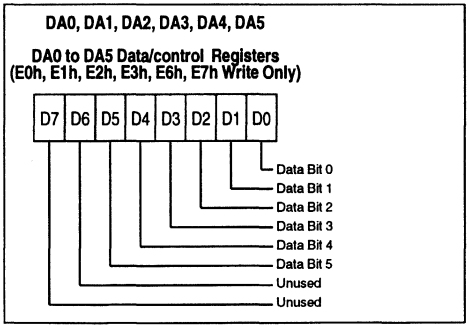
Each D/A converter of ST638x is composed by the following main blocks:

- pre-divider
- 6-bit counter
- data latches and compare circuits

The pre-divider uses the clock input frequency (8MHz typical) and its output clocks the 6-bit free-running counter. The data latched in the six registers (E0h, E1h, E2h, E3h, E6h and E7h) control the six D/A outputs (DA0,1,2, 3, 4 and 5). When all zeros are loaded the relevant output is an high logic level; all 1's correspond to a pulse with a 1/64 duty cycle and almost 100% zero level.

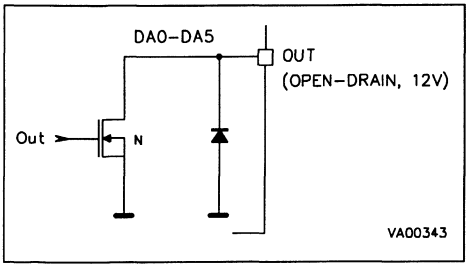
The repetition frequency is 31.25kHz and is related to the 8MHz clock frequency. Use of a different oscillator frequency will result in a different repetition frequency. All D/A outputs are open-drain with standard current drive capability and able to withstand up to 12V.

Figure 50. DA0-DA5 Data/Control Registers



DA0-DA5. These are the 6 bits of the PWM digital to analog converter. Undefined after reset.

Figure 51.6-bit PWM D/A Output Configuration



AFC A/D COMPARATOR

AFC A/D INPUT,IR/PC6 RESULT, VSYNC RESULT

The AFC macrocell contains an A/D comparator with five levels at intervals of 1V from 1V to 5V. The levels can all be lowered by 0.5V to effectively double the resolution.

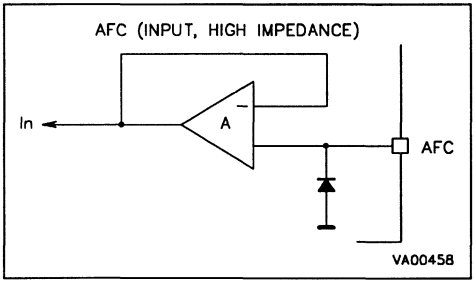
A/D Comparator

The A/D used to perform the AFC function (when high threshold is selected) has the following voltage levels: 1,2,3,4 and 5V. Bits 0-2 of AFC result register (E4h address) will provide the result in binary form (less than 1V is 000, greater than 5V is 101).

If the application requires a greater resolution, the sensitivity can be doubled by clearing to zero bit 2 of the OUTPUTS control register, address E5h. In this case all levels are shifted lower by 0.5V. If the two results are now added within a software routine then the A/D S-curve can be located within a resolution of 0.5V.

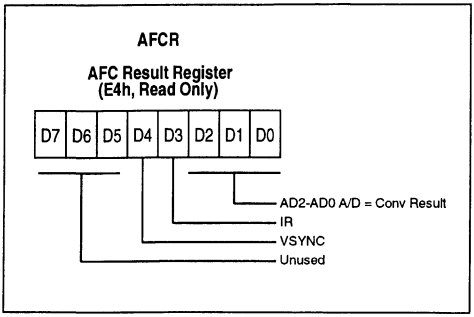
The A/D input has high impedance able to withstand up to 13V signals (input level tolerances ±200µV absolute and ± 100mV relative to 5V).

Figure 52. AFC Inputs Configuration Diagram



AFC A/D COMPARATOR (Continued)

Figure 53. AFC, IR and OSD Result Register



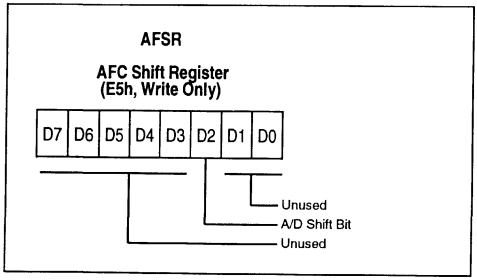
D7-D5. These bits are not used.

VSYNC. This bit reads the status of the VSYNC pin. It is inverted with respect to the pin.

IR. This bit reads the status of the IR latch. If a signal has been latched this bit will be high.

AD2-AD0. These bits store the real time conversion of the value present on the AFC input pin. Undefined reset value.

Figure 54. AFC Shift Register



D7, D6, D5, D4, D3, D1, D0. These bits are not used.

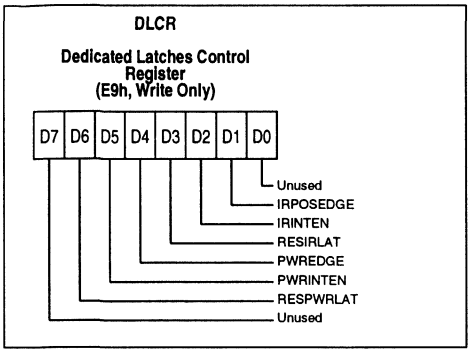
A/D Shift. This bit determines the voltage range of the AFC input. Writing a zero will select the 0.5V to 4.5V range. Writing a one will select the 1.0V to 5.0V range. Undefined after reset.

DEDICATED LATCHES

Two latches are available which may generate interrupts to the ST638x core. The IR latch is set either by the falling or rising edge of the signal on pin PC6(IRIN). If bit 1 (IRPOSEDGE) of the latches register (E9h) is high, then the latch will be triggered on the rising edge of the signal at PC6(IRIN). If bit 1 (IRPOSEDGE) is low, then the latch will be triggered on the falling edge of the signal at PC6(IRIN). The IR latch can be reset by setting bit 3 (RESIRLAT) of the latches register; the bit is set only and a high should be written every time the IR latch needs to be reset. If bit 2 (IRINTEN) of the latches register (E9h) is high, then the output of the IR latch, IRINTN, may generate an interrupt (#0). IRINTN is inverted with respect to the state of the IR latch. If bit 2 (IRINTEN) is low, then the output of the IR latch, IRINTN, is forced high. The state of the IR latch may be read from bit 3 (IRLATCH) of register E4h; if the IR latch is set, then bit 3 will be high. The PWR latch is set either by the falling or rising edge of the signal on pin PC4(PWRIN). If bit 4 (PWREDGE) of the latches register (E9h) is high, then the latch will be triggered on the rising edge of the signal at PC4(PWRIN). If bit 4 (PWREDGE) is low, then the latch will be triggered on the falling edge of the signal at PC4(PWRIN). The PWR latch can be reset by setting bit 6 (RESPWRLAT) of the latches register; the bit is set only and a high should be written every time the PWR latch needs to be reset. If bit 5 (PWRINTEN) of the latches register (E9h) is high, then the output of the PWR latch, PWRINTN, may generate an interrupt (#4). PWRINTN is inverted with respect to the state of the PWR latch. If bit 5 (PWRINTEN) is low, then the output of the PWR latch, PWRINTN, is forced high.

DEDICATED LATCHES(Continued)

Figure 55. Dedicated Latches Control Register



D7. This bit is not used

RESPWRLAT. Resets the PWR latch; this bit is set only.

PWRINTEN. This bit enables the PWRINT signal (#4) from the latch to the ST638x core. Undefined after reset.

PWREDGE. The bit determines the edge which will cause the PWRIN latch to be set. If this bit is high, than the PWRIN latch will be set on the rising edge of the PWRIN signal. Undefined after reset.

RESIRLAT. Resets the IR latch; this bit is set only.

IRINTEN. This bit enables the IRINTN signal (#0) from the latch to the ST638x core. Undefined after reset.

IRPOSEDGE. The bit determines the edge which will cause the IR latch to be set. If this bit is high, than the IR latch will be set on the rising edge of the IR signal. Undefined after reset.

D0. This bit is not used

ON-SCREEN DISPLAY (OSD)

The ST638x OSD macrocell is a CMOS LSI character generator which enable display of characters and symbols on the TV screen. The character rounding function enhances the readability of the characters. The ST638x OSD receives horizontal and vertical synchronization signal and outputs screen information via R, G, B and blanking pins. The main characteristics of the macrocell are listed below:

- Number of display characters: 5 lines by 15 columns.
- Number of character types: 128 characters in two banks of 64 characters. **Only one bank per screen can be used.**
- Character size: Four character heights (18H, 36H, 54H, 72H), two heights are available per screen, programmable by line.
- Character format: 6x9 dots with character rounding function.
- Character colour: Eight colours available programmable by word.
- Display position: 64 horizontal positions by 2/fosc and 63 vertical positions by 4H
- Word spacing: 64 positions programmable from 2/fosc to 128/fosc.
- Line spacing: 63 positions programmable from 4 to 252 H.
- Background: No background, square background or fringe background programmable by word.
- Background colour: Two of eight colours available programmable by word.
- Display output: Three character data output terminals (R,G,B) and a blank output terminal.
- Display on/off: Display data may be programmed on or off by word or entire screen. The entire screen may be blanked.

Format Specification

The entire display can be turned on or off through the use of the global enable bit or the display may be selectively turned on or off by word. To turn off the entire display, the global enable bit (GE) should be zero. If the global enable is one, the display is controlled by the word enable bits (WE). The global enable bit is located in the global enable register and the word enable bit is located in the space character preceding the word.

ON-SCREEN DISPLAY (Continued)

Each line must begin with a format character which describes the format of that line and of the first word. This character is not displayed.

A space character defines the format of subsequent words. A space character is denoted by a one in bit 6 in the display RAM. If bit 6 of the display RAM is a zero, the other six bits define one of the 64 display characters.

The colour, background and enable can be programmed by word. This information is encoded in the space character between words or in the format character at the beginning of each line. Five bits define the colour and background of the following word, and determine whether it will be displayed or not.

Characters are stored in a 6 x 9 dot format. One dot is defined vertically as 2H (horizontal lines) and horizontally as $2/f_{OSC}$ if the smallest character size is enabled. There is no space between characters or lines if the vertical space enable (VSE) and horizontal space enable (HSE) bits are both zero. This allows the use of special graphics characters.

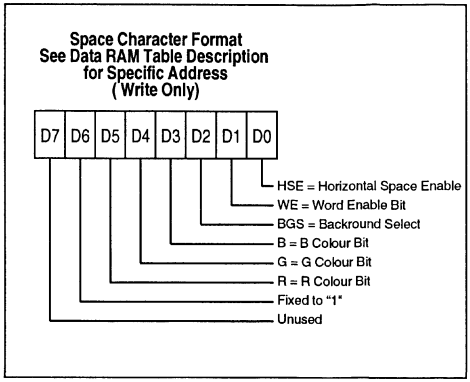
The normal alphanumeric character set is formatted to be 5 x 7 with one empty row at the top and one at the bottom and one empty column at the right. If VSE and HSE are both zero, then the spacing between alphanumeric characters is 1 dot and the spacing between lines of alphanumeric characters is 2H.

The character size is programmed by line through the use of the size bit (S) in the format character and the global size bits (GS1 and GS2). The vertical spacing enable bit (VSE) located in the format character controls the spacing between lines. If this bit is set to one, the spacing between lines is defined by the vertical spacing register, otherwise the spacing between lines is 0.

The spacing between words is controlled by the horizontal space enable bit (HSE) located in the space character. If this bit is set to one, the spacing between words is defined by the horizontal spacing register, otherwise the space character width of 6 dots is the spacing between words.

The formats for the display character, space character and format character are described hereafter.

Figure 56. Space Character Register Explanation



D7. Not used.

D6. This pin is fixed to "1".

R, G, B. *Colour.* The 3 colour control bits define the colour of the following word as shown in table below.

Space Character Register Colour Setting.

R	G	B	Colour
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

BGS. *Background Select.* The background select bit selects the desired background for the following word. There are two possible backgrounds defined by the bits in the Background Control Register.

"0" - The background on the following word is enabled by BG0 and the colour is set by R0, G0, and B0.

"1" - The background on the following word is enabled by BG1 and the colour is set by R1, G1, and B1.

ON-SCREEN DISPLAY (Continued)

WE. *Word Enable.* The word enable bit defines whether or not the following word is displayed.

"0" -The word is not displayed.

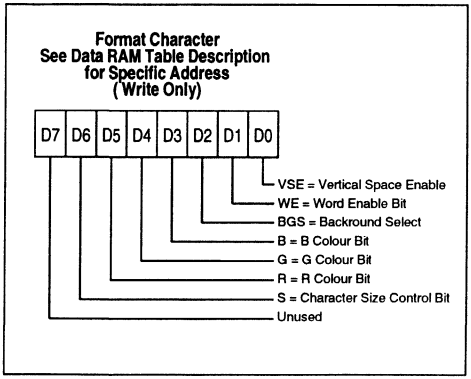
"1" -If the global enable bit is one, then the word is displayed.

HSE. *Horizontal Space Enable.* The horizontal space enable bit determines the spacing between words. The space between characters is always 0. The alphanumeric character set is implemented in a 5 x 7 format with one empty column to the right and one empty row above and below so that the space between alphanumeric characters will be one dot.

"0" -The space between words is equal to the width of the space character, which is 6 dots.

"1" -The space between words is defined by the value in the horizontal space register plus the width of the space character.

Figure 57. Format Character Register Explanation



D7. This bit is not used

S. *Character Size.* The character size bit, along with the global size bits (GS2 and GS1) located in the horizontal space register, specify the character size for each line as defined in Table 14.

R, G, B. *Colour.* The 3 colour control bits define the colour of the following word as shown in Table 13.

BGS. *Background Select.* The background select bit selects the desired background for the following word. There are two possible backgrounds defined by the bits in the Background Control Register.

"0" -The background on the following word is enabled by BG0 and the colour is set by R0, G0, and B0.

"1" -The background on the following word is enabled by BG1 and the colour is set by R1, G1, and B1.

WE. *Word Enable.* The word enable bit defines whether or not the following word is displayed.

"0" -The word is not displayed.

"1" -If the global enable bit is one, then the word is displayed.

VSE. *Vertical Space Enable.* The vertical space enable bit determines the spacing between lines.

"0" -The space between lines is equal to 0H. The alphanumeric character set is implemented in a 5 x 7 format with one empty column to the right and one empty row above and one below and stored in a 6 x 9 format.

"1" -The space between lines is defined by the value in the vertical space register.

Table 13. Format Character Register Colour Setting.

R	G	B	Colour
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

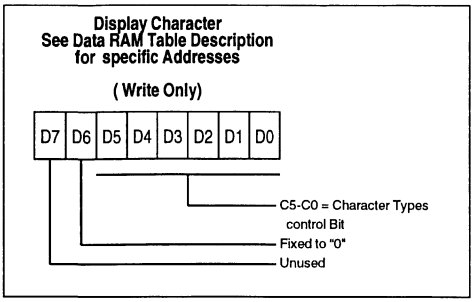
Table 14. Format Character Register Size Setting

GS2	GS1	S	Vertical Height	Horizontal length
0	0	0	18H	6 TDOT
0	0	1	36H	12 TDOT
0	1	0	18H	6 TDOT
0	1	1	54H	18 TDOT
1	0	0	36H	12 TDOT
1	0	1	54H	18 TDOT
1	1	0	36H	12 TDOT
1	1	1	72H	24 TDOT

TDOT= $2/f_{OSC}$

ON-SCREEN DISPLAY (Continued)

Figure 58. Display Character Register Explanation



D7. This bit is not used.

D6. This bit is fixed to "0".

C5-C0. *Character type.* The 6 character type bits define one of the 64 available character types. These character types are shown on the following pages.

Character Types

The character set is user defined as ROM mask option.

Register and RAM Addressing

The OSD contains seven registers and 80 RAM locations. The seven registers are the Vertical Start Address register, Horizontal Start Address register, Vertical Space register, Horizontal Space register, Background Control register, Global Enable register and Character Bank Select register. The Global Enable register can be written at any time by the ST638x Core. The other six registers and the RAM can only be read or written to if the global enable is zero.

The six registers and the RAM are located on two pages of the paged memory of the ST638x MCUs; the Character Bank Select register is located outside the paged memory at address EDh. Each page contains 64 memory locations. This paged memory is at memory locations 00h to 3Fh in the ST638x memory map. A page of memory is enabled by setting the desired page bit, located in the Data Ram Bank Register, to a one. The page register is location E8h. A one in bit five selects page 5, located on the OSD and a one in bit 6 selects page 6 on the OSD. Table 15 shows the addresses of the OSD registers and RAM.

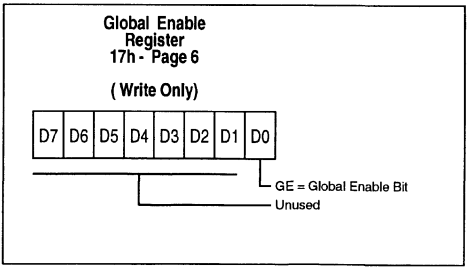
Table 15. OSD Control Registers and Data RAM Addressing

Page	Address	Register or RAM
5	00h - 3Fh	RAM Locations 00h - 3Fh
6	00h - 0Fh	RAM Locations 00h - 0Fh
6	10h	Vertical Start Register
6	11h	Horizontal Start Register
6	12h	Vertical Space Register
6	13h	Horizontal Space Register
6	14h	Background Control Register
6	17h	Global Enable Register
No Page	EDh	Character Bank Select Register

OSD Global Enable Register

This register contains the global enable bit (GE). It is the only register that can be written at any time regardless of the state of the GE bit. It is a write only register.

Figure 59. Global Enable Bit



D7-D1. These bits are not used

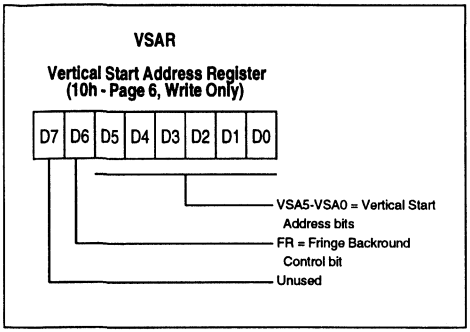
GE. *Global Enable.* This bit allows the entire display to be turned off.

"0" - The entire display is disabled. The RAM and other registers of the OSD can be accessed by the Core.

"1" - Display of words is controlled by the word enable bits (WE) located in the format or space character. The other registers and RAM cannot be accessed by the Core.

ON-SCREEN DISPLAY (Continued)

Figure 60. Vertical Start Address Register



D7. This bit is not used

FR. *Fringe Background.* This bit changes the background from a box background to a fringe background. The background is enabled by word as defined by either BG0 or BG1.

"0" -The background is defined to be a box which is 7 x 9 dots.

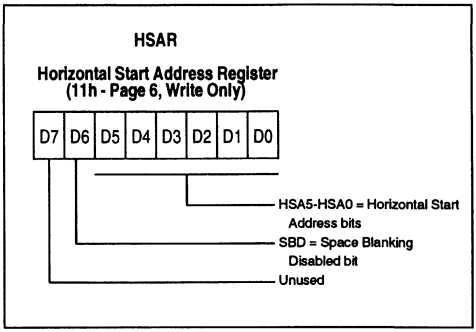
"1" -The background is defined to be a fringe.

VSA5-VSA0. Vertical Start Address. These bits determine the start position of the first line in the vertical direction. The 6 bits can specify 63 display start positions of interval 4H. The first start position will be the fourth line of the display. The vertical start address is defined VSA0 by the following formula.

Vertical Start Address = $4H(2^5(VSA5) + 2^4(VSA4) + 2^3(VSA3) + 2^2(VSA2) + 2^1(VSA1) + 2^0(VSA0))$

The case of all Vertical Start Address bits being zero is illegal.

Figure 61. Horizontal Start Address Register



D7. This bit is not used.

SBD. *Space Blanking Disable.* This bit controls whether or not the background is displayed when outputting spaces. If two background colours are used on adjacent words, then the background should not be displayed on spaces in order to make a nice break between colours. If an even background around an area of text is desired, as in a menu, then the background should be displayed when outputting spaces.

"0" -The background during spaces is controlled by the background enable bits (BG0 and BG1) located in the Background Control register.

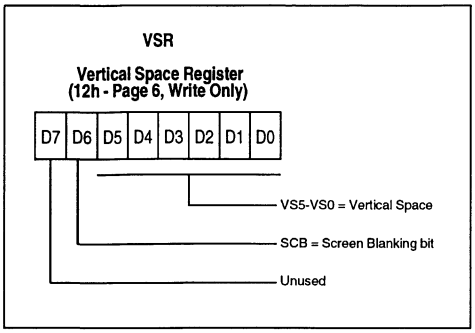
"1" -The background is not displayed when outputting spaces.

HSA5, HSA0 - *Horizontal Start Address bits.* These bits determine the start position of the first character in the horizontal direction. The 6 bits can specify 64 display start positions of interval $2/f_{OSC}$ or 400ns. The first start position will be at 4.0µs because of the time needed to access RAM and ROM before the first character can be displayed. The horizontal start address is defined by the following formula.

Horizontal Start Address = $2/f_{OSC}(10.0 + 2^5(HSA5) + 2^4(HSA4) + 2^3(HSA3) + 2^2(HSA2) + 2^1(HSA1) + 2^0(HSA0))$

ON-SCREEN DISPLAY (Continued)

Figure 62. Vertical Space Register



D7. This bit is not used

SCB. *Screen Blanking.* This bit allows the entire screen to be blanked.

"0" -The blanking output signal (VBLK) is active only when displaying characters.

"1" -The blanking output signal (VBLK) is always active. Characters in the display RAM are still displayed.

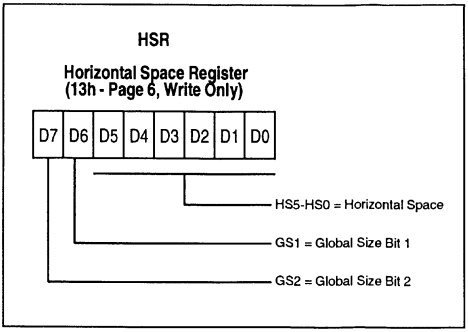
When this bit is set to one, the screen is blanked also without setting the Global Enable bit to one (OSD disabled).

VS5 , VS0. *Vertical Space.* These bits determine the spacing between lines if the Vertical Space Enable bit (VSE) in the format character is one. If VSE is zero there will be no spaces between lines. The Vertical Space bits can specify one of 63 spacing values from 4H to 252H. The space between lines is defined by the following formula.

Space between lines = $4H(2^5(VS5) + 2^4(VS4) + 2^3(VS3) + 2^2(VS2) + 2^1(VS1) + 2^0(VS0))$

The case of all Vertical Start Address bits being zero is illegal.

Figure 63. Horizontal Space Register



GS2,GS1. *Global Size.* These bits along with the size bit (S) located in the Character format word specify the character size for each line as defined in table 16.

Table 16. Horizontal Space Register Size Setting.

GS2	GS1	S	Vertical Height	Horizontal Length
0	0	0	18H	6 TDOT
0	0	1	36H	12 TDOT
0	1	0	18H	6 TDOT
0	1	1	54H	18 TDOT
1	0	0	36H	12 TDOT
1	0	1	54H	18 TDOT
1	1	0	36H	12 TDOT
1	1	1	72H	24 TDOT

Note: TDOT= 2/fOSC

HS5, HS0 . *Horizontal Space* . These bits determine the spacing between words if the Horizontal Space Enable bit (HSE) located in the space character is a one. The space between words is then equal to the width of the space character plus the number of tdots specified by the Horizontal Space bits. The 6 bits can specify one of 64 spacing values ranging from 2/fosc to 128/fosc. The formula is shown below for the smallest size character(18H). If larger size characters are being displayed the spacing between words will increase proportionately. Multiply the value below by 2, 3 or 4 for character sizes of 36H, 54H and 72H respectively.

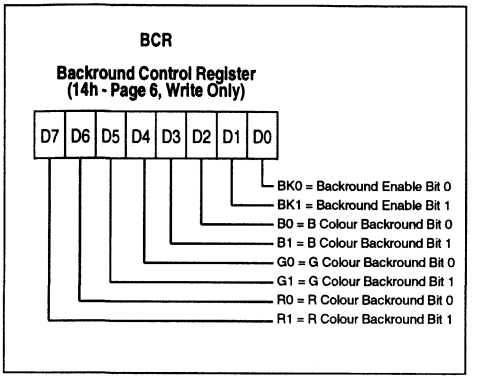
Space between words (not including the space character)$=2/f_{OSC}(1+2^5(HS5)+2^4(HS4)+2^3(HS3)+2^2(HS2)+2^1(HS1)+2^0(HS0))$

ON-SCREEN DISPLAY (Continued)

Background Control Register

This register sets up two possible backgrounds. The background select bit (BGS) in the format or space character will determine which background is selected for the current word.

Figure 64. Backround Control Register

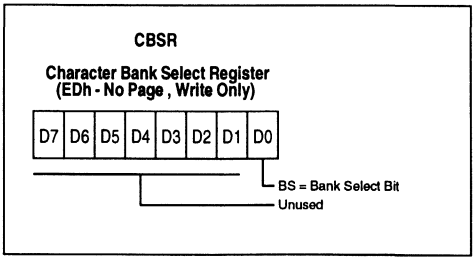


R1,R0,G1,G0,B1,B0. Background Colour. These bits define the colour of the specified background, either background 1 or background 0 as defined in table below.

Table 17. Background Register Colour Setting.

RX	GX	BX	Colour
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

Figure 65. Character Bank Select Register



BK1,BK0. *Background Enable.*These bits determine if the specified background is enabled or not.

"0" -The following word does not have a background.

"1" -There is a background around the following word.

D7-D1. These bits are not used

BS. *Bank Select.* This bit select the character bank to be used. The lower bank is selected with 0. The value can be modified only when the OSD is OFF (GE=0). No reset value.

OSD Data RAM

The contents of the data RAM can be accessed by the ST638x MCUs only when the global enable bit (GE) in the Global Enable register is a zero.

The first character in every line is the format character. This character is not displayed. It defines the size of the characters in the line and contains the vertical space enable bit. This character also defines the colour, background and display enable for the first word in the line. Subsequent characters are either spaces or one of the 64 available character types.

The space character defines the colour, background, display enable and horizontal space enable for the following word. Since there are 5 display lines of 15 characters each, the display RAM must contain 5 lines x (15 characters + 1 format character) or 80 locations. The RAM size is 80 locations x 7 bits. The data RAM map is shown inTable 18.

ON-SCREEN DISPLAY (Continued)

Table 12. OSD RAM Map

Column				0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A0				0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
A1				0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
A2				0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
A3				0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Page	A5	A4	LINE																
5	0	0	1	FT	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch
5	0	1	2	FT	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch
5	1	0	3	FT	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch
5	1	1	4	FT	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch
6	0	0	5	FT	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch
				AVAILABLE SCREEN SPACE															

Notes: FT. The format character required for each line. Characters in columns 1 thru 15 are displayed.

Ch. (Byte) Character (Index into OSD character generator) or space character

Emulator Remarks

There are a few differences between emulator and silicon. For noise reasons, the OSD oscillator pins are not available: the internal oscillator cannot be disabled and replaced by an external coil. In the emulator, the Character Bank Select register can be written also with Global Enable bit set, while this is not allowed in the device.

Application Notes

1 - The OSD character generator is composed of a dual port video ram and some circuitry. It needs two input signals VSYNC and HSYNC to syncronize its dedicated oscillator to the TV picture. It generates 4 output signals, that can be used from the TV set to generate the characters on the screen. For instance, they can be used to feed the SCART plug, providing an adequate buffer to drive the low impedance (75Ω) of the SCART inputs.

2 - The Core sees the OSD as a number of RAM locations (80) plus a certain number of control registers (6). These 86 locations are mapped in two pages of the dynamic data ram address range (0h..3Fh).

In page 5 (load 20h in the register 0E8h), there are 64 bytes of RAM, the ones of the first 4 rows (16 bytes each row, 15 characters per row maximum, plus an hidden leading format character). In page 6 (load 40h in register 0E8h), the 16 bytes of the fifth row (0..0Fh), and the 6 control registers (10h..14h,17h).

3 - The video RAM is a dual port ram. That means that it can be addressed either from the Core or from the OSD circuitry itself. To reduce the complexity of the circuitry, and thus its cost, some restrictions have been introduced in the use of the OSD.

a. The Core can Only write to any of the 86 locations (either video RAM or control registers).

b. The Core can Only write to any of the leading 85 locations when the OSD oscillator is OFF. Only the last location (control register 17h in page 6) can be addressed at any time. This is the Global Enable Register, which contains only the GE bit. If it is set, the OSD is on, if it is reset the OSD is off.

4 - The timing of the on/off switching of the OSD oscillator is the following:

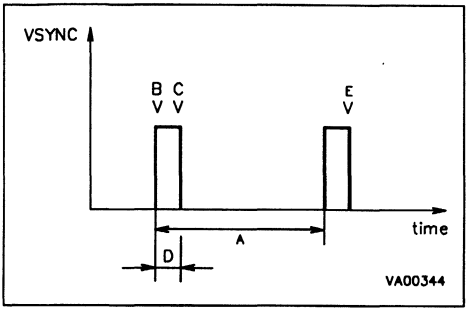
a. GE bit is set. The OSD oscillator will start on the next VSYNC signal.

b. GE bit is reset. The OSD oscillator will be immediately switched off.

ON-SCREEN DISPLAY (Continued)

To avoid a bad visual impression, it is important that the GE bit is set before the end of the flyback time when changing characters. This can be done inside the VSYNC interrupt routine. The following diagram can explain better:

Figure 66. OSD Oscillator ON/OFF Timing



Notes: A - Picture time: 20 mS in PAL/SECAM.
B - VSYNC interrupt, if enabled.
C - Starting of OSD oscillator, if GE = 1.
D - Flyback time.

When modifying the picture display (i.e.: a bar graph for an analog control), it is important that the switching on of the GE bit is done before the the end of the flyback time (D in Figure 69). If the GE bit is set after the end of the flyback time then the OSD will not start until the begining of the next frame. This results in one frame being lost and will result in a Flicker on the screen. One method to be sure to avoid the flicker is to wait for the VSYNC interrupt at the start of the flyback; once the VSYNC interrupt is detected, then the GE bit can be set to zero, the characters changed, and the GE set to one. All this should occur before the end of the flyback time in order not to lose a frame. The correct edge of the interrupt must be chosen.

The VSYNC pin may alternatively be sampled by software in order to know the status; this can be done by reading bit 4 of register E4h; this bit is inverted with respect to the VSYNC pin.

6 - An OSD end of line Bar is present in the ST63P8X piggyback and ST638x ROM, EPROM and OTP devices when using the background mode. If this bar is present with software running in the piggybacks then it is also present on the ROM mask version. If the end of line bar is seen to be eliminated by software in the piggyback, then it is also be eliminated in the ROM mask version.

The bar appears at the end of the line in the background mode when the last character is a space character, the first format character is defined with S=0 (size 0)and the backround is not displayed during the space. The bar is the colour of the background defined by the space character. To eliminate the bar:

a. If two backgrounds are used then the bar should be moved off the screen by using large word spaces instead of character spaces. If there are not enough spaces before the end of the line, then the location of the valid characters should be moved so they appear at the end of the line (and hence no bar); positioning can be compensated using the horizontal start register.

b. If only one background is used, then the other background should be transparent in order to eliminate the bar.

7 - The OSD oscillator external network should consist of a capacitor on each of the OSD oscillator pins to ground together with an inductance between pins. The user should select the two capacitors to be the same value (15pF to 25pF each is recommended). The inductance is chosen to give the desired OSD oscillator frequency for the application (typically 56µH).

SOFTWARE DESCRIPTION

The ST63xx software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum; in short to provide byte efficient programming capability. The ST63xx Core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

Addressing Modes

The ST63xx Core has nine addressing modes which are described in the following paragraphs. The ST63xx Core uses three different address spaces : Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte that is processed by the instruction is stored in the location that follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The Core can address the four RAM registers X,Y,V,W (locations 80h, 81h, 82h, 83h) in the short-direct addressing mode. In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) that use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction that follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits that characterize the kind of the test, one bit that determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits that give the span of the branch (0h to Fh) that must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80h,81h). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

SOFTWARE DESCRIPTION (Continued)

Instruction Set

The ST63xx Core has a set of 40 basic instructions. When these instructions are combined with nine addressing modes, 244 usable opcodes can be obtained. They can be divided into six different types:load/store, arithmetic/logic, conditional branch, control instructions, jump/call, bit manipulation. The following paragraphs describe the different types.

All the instructions within a given type are presented in individual tables.

Load & Store. These instructions use one,two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data. See Table 13.

Table 13. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
#. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
*. Not Affected

SOFTWARE DESCRIPTION (Continued)

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator. See Table 14.

Table 14. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	*
AND A, (Y)	Indirect	1	4	Δ	*
AND A, rr	Direct	2	4	Δ	*
ANDI A, #N	Immediate	2	4	Δ	*
CLR A	Short Direct	2	4	Δ	Δ
CLR rr	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
#. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
*. Not Affected

SOFTWARE DESCRIPTION (Continued)

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met. See Table 15.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations. See Table 16.

Control Instructions. The control instructions control the MCU operations during program execution. See Table 17.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space. Refer to Table 18.

Table 15. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ. Affected
*. Not Affected

Table 16. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
*. Not Affected

Table 17. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated and a WAIT is automatically executed instead of a STOP if the hardware activated watchdog function is selected.
Δ. Affected
*. Not Affected

Table 18. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc. 12-bit address;
*. Not Affected

SOFTWARE DESCRIPTION (Continued)

Opcode Map Summary. The following table contains an opcode map for the instructions used on the MCU.

Low / Hi	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Low / Hi
0 0000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 LD a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 pcr	4 LD a,(y) 1 ind	0 0000
1 0001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC x 1 sd	2 JRC e 1 prc	4 LDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 JRZ e 1 pcr	4 DEC x 1 sd	2 JRC e 1 pcr	4 LD a,rr 2 dir	1 0001
2 0010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b4,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 CP a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b4,rr 2 b.d	2 JRZ e 1 pcr	4 COM a 1 inh	2 JRC e 1 pcr	4 CP a,(y) 1 ind	2 0010
3 0011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b4,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,x 1 sd	2 JRC e 1 prc	4 CPI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b4,rr 2 b.d.	2 JRZ e 1 pcr	4 LD x,a 1 sd	2 JRC e 1 pcr	4 CP a,rr 2 dir	3 0011
4 0100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b2,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 ADD a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b2,rr 2 b.d	2 JRZ e 1 pcr	2 RETI 1 inh	2 JRC e 1 pcr	4 ADD a,(y) 1 ind	4 0100
5 0101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b2,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC y 1 sd	2 JRC e 1 prc	4 ADDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b2,rr 2 b.d	2 JRZ e 1 pcr	4 DEC y 1 sd	2 JRC e 1 pcr	4 ADD a,rr 2 dir	5 0101
6 0110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b6,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 INC (x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b6,rr 2 b.d	2 JRZ e 1 pcr	2 STOP 1 inh	2 JRC e 1 pcr	4 INC (y) 1 ind	6 0110
7 0111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b6,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,y 1 sd	2 JRC e 1 prc	# 	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b6,rr 2 b.d	2 JRZ e 1 pcr	4 LD y,a 1 sd	2 JRC e 1 pcr	4 INC rr 2 dir	7 0111
8 1000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b1,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 LD (x),a 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b1,rr 2 b.d	2 JRZ e 1 pcr	# 	2 JRC e 1 pcr	4 LD (y),a 1 ind	8 1000
9 1001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b1,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC v 1 sd	2 JRC e 1 prc	# 	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b1,rr 2 b.d	2 JRZ e 1 pcr	4 DEC v 1 sd	2 JRC e 1 pcr	4 LD rr,a 2 dir	9 1001
A 1010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b5,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 AND a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b5,rr 2 b.d	2 JRZ e 1 pcr	4 RLC a 1 inh	2 JRC e 1 pcr	4 AND a,(y) 1 ind	A 1010
B 1011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b5,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,v 1 sd	2 JRC e 1 prc	4 ANDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b5,rr 2 b.d	2 JRZ e 1 pcr	4 LD v,a 1 sd	2 JRC e 1 pcr	4 AND a,rr 2 dir	B 1011
C 1100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b3,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 SUB a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b3,rr 2 b.d	2 JRZ e 1 pcr	2 RET 1 inh	2 JRC e 1 pcr	4 SUB a,(y) 1 ind	C 1100
D 1101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b3,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC w 1 sd	2 JRC e 1 prc	4 SUBI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b3,rr 2 b.d	2 JRZ e 1 pcr	4 DEC w 1 sd	2 JRC e 1 pcr	4 SUB a,rr 2 dir	D 1101
E 1110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b7,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 DEC (x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b7,rr 2 b.d	2 JRZ e 1 pcr	2 WAIT 1 inh	2 JRC e 1 pcr	4 DEC (y) 1 ind	E 1110
F 1111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b7,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,w 1 sd	2 JRC e 1 prc	# 	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b7,rr 2 b.d	2 JRZ e 1 pcr	4 LD w,a 1 sd	2 JRC e 1 pcr	4 DEC rr 2 dir	F 1111

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:
# Indicates Illegal Instructions
e 5 Bit Displacement
b 3 Bit Address
rr1byte dataspace address
nn 1 byte immediate data
abc 12 bit address
ee 8 bit Displacement

ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from :

$$Tj = T_A + PD \times RthJA$$

Where : T_A = Ambient Temperature.

RthJA = Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint = $I_{DD} \times V_{DD}$ (chip internal power).

Pport = Port power dissipation (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage (AFC IN)	V_{SS} - 0.3 to +13	V
V_I	Input Voltage (Other Inputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
V_O	Output Voltage (PA4-PA7, PC4-PC7, DA0-DA5)	V_{SS} - 0.3 to +13	V
V_O	Output Voltage (Other Outputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
I_O	Current Drain per Pin Excluding V_{DD}, V_{SS}, PA6, PA7	± 10	mA
I_O	Current Drain per Pin (PA6, PA7)	± 50	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	150	mA
Tj	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Note : Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device . This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PSDIP42			67	°C/W

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	1 Suffix Version	0		70	°C
V_{DD}	Operating Supply Voltage		4.5	5.0	6.0	V
f_{OSC}	Oscillator Frequency RUN & WAIT Modes			8	8.1	MHz
f_{OSDOSC}	On-screen Display Oscillator Frequency				8.0	MHz

EEPROM INFORMATION

The ST63xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = 0 to +70°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	All I/O Pins			0.2xV_{DD}	V
V_{IH}	Input High Level Voltage	All I/O Pins	0.8xV_{DD}			V
V_{HYS}	Hysteresis Voltage(1)	All I/O Pins V_{DD} = 5V		1.0		V
V_{OL}	Low Level Output Voltage	DA0-DA5, PB0-PB6, OSD Outputs, PC0-PC7, O0, O1, PA0-PA5 V_{DD} = 4.5V I_{OL} = 1.6mA I_{OL} = 5.0mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	PA6-PA7 V_{DD} = 4.5V I_{OL}= 1.6mA I_{OL}= 25mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	OSDOSCout, OSCout V_{DD} = 4.5V I_{OL}= 0.4mA			0.4	V
V_{OL}	Low Level Output Voltage	VS Output V_{DD} = 4.5V I_{OL}= 0.5mA I_{OL}= 1.6mA			 0.4 1.0	 V V
V_{OH}	High Level Output Voltage	PB0-PB7, PA0-PA3, OSD Outputs V_{DD} = 4.5V I_{OH} = – 1.6mA	4.1			V
V_{OH}	High Level Output Voltage	OSDOSCout, OSCout, V_{DD} = 4.5V I_{OH}= – 0.4mA	4.1			V
V_{OH}	High Level Output Voltage	VS Output V_{DD} = 4.5V I_{OH}= – 0.5mA	4.1			V
I_{PU}	Input Pull Up Current Input Mode with Pull-up	PB0-PB6, PA0-PA3, PC0-PC3 V_{IN}= V_{SS}	– 100	– 50	– 25	mA
I_{IL} I_{IH}	Input Leakage Current	OSCin V_{IN}= V_{SS} V_{IN}= V_{DD}	– 10 0.1	– 1 1	– 0.1 10	µA
I_{IL}	Input Pull-down current in Reset	OSCin	100			µA
I_{IL} I_{IH}	Input Leakage Current	All I/O Input Mode no Pull-up OSDOSCin V_{IN}= V_{DD} or V_{SS}	– 10		10	µA

DC ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{DD}RAM	RAM Retention Voltage in RESET		1.5			V
I_{IL} I_{IH}	Input Leakage Current	Reset Pin with Pull-up V_{IN}= Vss	– 50	– 30	– 10	µA
I_{IL} I_{IH}	Input Leakage Current	AFC Pin V_{IH}= V_{DD} V_{IL}= Vss V_{IH}= 12.0V	– 1		1 40	µA
I_{OH}	Output Leakage Current	DA0-DA5, PA4-PA5, PC0-PC7, O0, O1 V_{OH} = V_{DD}			10	µA
I_{OH}	Output Leakage Current High Voltage	DA0-DA5, PA4-PA7, PC4-PC7, O0, O1 V_{OH} = 12V			40	µA
I_{DD}	Supply Current RUN Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6.0V		6	16	mA
I_{DD}	Supply Current WAIT Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6V		3	10	mA
I_{DD}	Supply Current at transition to RESET	f_{OSC}= Not App, ILoad= 0mA V_{DD}= 6V		0.1	1	mA
V_{ON}	Reset Trigger Level ON	RESET Pin			0.3xV_{DD}	V
V_{OFF}	Reset Trigger Level OFF	RESET Pin	0.8xV_{DD}			V
V_{TA}	Input Level Absolute Tolerance	A/D AFC Pin V_{DD} = 5V			±200	mV
V_{TR}	Input Level Relatice Tolerance (1)	A/D AFC Pin Relative to other levels V_{DD} = 5V			±100	mV

Note: 1. Not 100% Tested

AC ELECTRICAL CHARACTERISTICS

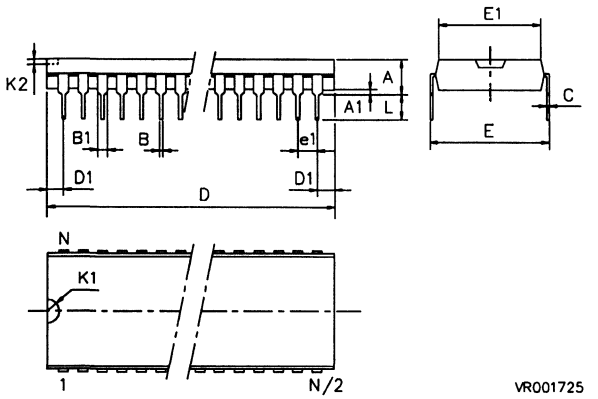
(T_A = 0 to +70°C, f_{OSC}=8MHz, V_{DD}=4.5 to 6.0V unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{WRES}	Minimum Pulse Width	RESET Pin	125			ns
tO_{HL}	High to Low Transition Time	PA6, PA7 V_{DD} = 5V, CL = 1000pF (2)		100		ns
tO_{HL}	High to Low Transition Time	DA0-DA5, PB0-PB6, OSD Outputs, PC0-PC7, V_{DD} = 5V, CL = 100pF		20		ns
tO_{LH}	Low to High Transition Time	PB0-PB6, PA0-PA3, OSD Outputs, PC0-PC3 V_{DD} = 5V, CL = 100pF		20		ns
tO_H	Data HOLD Time SPI after clock goes low I^2CBUS/S-BUS Only		175			ns
f_{DA}	D/A Converter Repetition Frequency(1)			31.25		kHz
f_{SIO}	SIO Baud Rate(1)			62.50		kHz
t_{WEE}	EEPROM Write Time	T_A = 25°C One Byte		5	10	ms
Endurance	EEPROM WRITE/ERASE Cycles	Q_A L_{OT} Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention (4)	T_A = 25°C	10			years
C_{IN}	Input Capacitance (3)	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance (3)	All outputs Pins			10	pF
COSCin, COSCout	Oscillator Pins Internal Capacitance(3)			5		pF
COSDin, COSDout	OSD Oscillator External Capacitance	Recommended	15		25	pF

Notes:
1. A clock other than 8 MHz will affect the frequency response of those peripherals (D/A, 62.5kHz and SPI) whose clock is derived from the system clock.
2. The rise and fall times of PORT A have been reduced in order to avoid current spikes while maintaining a high drive capability
3. Not 100% Tested
4. Based on extrapolated data

PACKAGE MECHANICAL DATA

Figure 67. ST638x 42 Pin Plastic Dual-In-line Package



Dim.	mm			Inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D	36.32		39.12	1.430		1.540
D1	–	–	–	–	–	–
E			18.54			0.730
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	42					

ORDERING INFORMATION

The following chapter deals with the procedure for transfer the Program/Data ROM codes to SGS-THOMSON.

Communication of the ROM Codes. To communicate the contents of Program /Data ROM memories to SGS-THOMSON, the customer has to send a 5" Diskette with:

- one file in INTEL INTELLEC 8/MDS FORMAT for the PROGRAM Memory
- one file in INTEL INTELLEC 8/MDS FORMAT for the ODD and EVEN ODD OSD Characters
- one file in INTEL INTELLEC 8/MDS FORMAT for the EEPROM initial content (this file is optional)
- a filled Option List form as described in the OPTION LIST paragraph.

The program ROM should respect the ROM Memory Map as in Table 19.

The ROM code must be generated with ST6 assembler. Before programming the EPROM, the buffer of the EPROM programmer must be filled with FFh.

Customer EEPROM Initial Contents: Format

a. The content should be written into an INTEL INTELLEC format file.

b. In the case of 384 bytes of EEPROM, the starting address is 000h

and the end address is 7Fh. The order of the pages (64 bytes each) is an in the specification (ie. b7, b1 b0: 001, 010, 011, 101, 110. 111).

c. Undefined or don't care bytes should have the content FFh.

OSD Test Character. IN ORDER TO ALLOW THE TESTING OF THE ON-CHIP OSD MACROCELL THE FOLLOWING CHARACTER MUST BE PROVIDED AT THE FIXED 3Fh (63) POSITION OF THE SECOND OSD BANK.

Listing Generation & Verification. When SGS-THOMSON receives the files, a computer listing is generated from them. This listing refers extractly to the mask that will be used to produce the microcontroller. Then the listing is returned to the customer that must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed list constitutes a part of the contractual agreement for the creation of the customer mask. SGS-THOMSON sales organization will provide detailed information on contractual points.

Figure 68. OSD Test Character

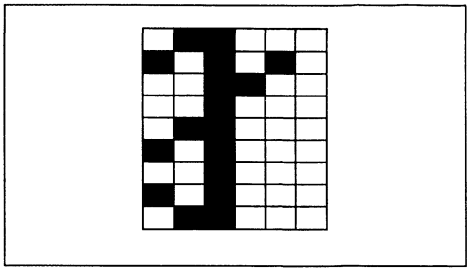


Table 19. ROM Memory Map

ROM Page	Device Address	EPROM Address (1)	Description
Page 0	0000h-007Fh 0080h-07FFh	0000h-007Fh 0080h-07FFh	Reserved User ROM
Page 1 "STATIC"	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
Page 2	0000h-000Fh 0010h-07FFh	1000h-100Fh 1010h-17FFh	Reserved User ROM
PAGE 3	0000h-000Fh 0010h-07FFh	1800h-180Fh 1810h-1FFFh	Reserved user ROM (End of 8K)
Page 4	0000h-000Fh 0010h-07FFh	2000h-200Fh 2010h-27FFh	Reserved User ROM
Page 5	0000h-000Fh 0010h-07FFh	2800h-280Fh 2810h-2FFFh	Reserved User ROM
Page 6	0000h-000Fh 0010h-07FFh	3000h-300Fh 3010h-37FFh	Reserved User ROM (End of 14K)
Page 7	0000h-000Fh 0010h-07FFh	3800h-380Fh 3810h-3FFFh	Reserved User ROM
Page 8	0000h-000Fh 0010h-07FFh	4000h-400Fh 4010h-47FFh	Reserved User ROM
Page 9	0000h-000Fh 0010h-07FFh	4800h-480Fh 4810h-4FFFh	Reserved User ROM (End of 20K)

Notes:

1. EPROM addresses are related to the use of ST63P8X piggyback emulation devices.

ORDERING INFORMATION TABLE

Sales Type	ROM/EEPROM Size	D/A Converter	Temperature Range	Package
ST6365B1/XX	8K/384 Bytes	4	0 to + 70 °C	PSDIP42
ST6367B1/XX	8K/384 Bytes	6	0 to + 70 °C	PSDIP42
ST6375B1/XX	14K/384 Bytes	4	0 to + 70 °C	PSDIP42
ST6377B1/XX	14K/384 Bytes	6	0 to + 70 °C	PSDIP42
ST6385B1/XX	20K/384 Bytes	4	0 to + 70 °C	PSDIP42
ST6387B1/XX	20K/384 Bytes	6	0 to + 70 °C	PSDIP42

Note: "XX" Is the ROM code identifier that is allocated by SGS-THOMSON after receipt of all required options and the related ROM file.

ST636x, 7x, 8x MICROCONTROLLER OPTION LIST

Customer: . . . . . .
Address: . . . . . .
Contact: . . . . . .
Phone No: . . . . . .
Reference: . . . . . .

Device [ ] (d) Package [ ] (p) Temperature Range [ ] (t)
For marking one line with 10 characters maximum is possible

Special Marking [ ] (y/n) Line1 " _ _ _ _ _ _ _ _ _ _ " (N)

Notes:
(d) 1= ST6365, 2 = ST6367, 3 = ST6375, 4 = ST6377, 5 = ST6385, 6 = ST6387
(p) B= Dual in Line Plastic
(t) 1= 0 to 70°C

(N) Letters, digits, ' . ', ' - ', ' / ' and spaces only

Marking: the default marking is equivalent to the sales type only (part number).

OSD POLARITY OPTIONS (Put a cross on selected item) :

	POSITIVE	NEGATIVE
VSYNC,HSYNC	[ ]	[ ]
R,G,B	[ ]	[ ]
BLANK	[ ]	[ ]

CHECK LIST:

	YES	NO
ROM CODE	[ ]	[ ]
OSD Code: ODD & EVEN	[ ]	[ ]
EEPROM Code (if Desired)	[ ]	[ ]

Signature .....

Date .....

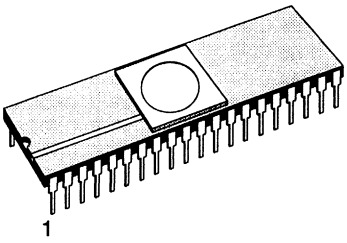
SGS-THOMSON MICROELECTRONICS

ST63E85, T85
ST63E87, T87

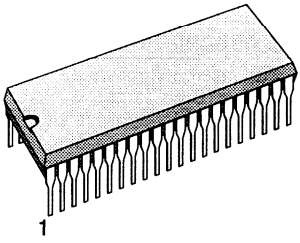
8-BIT EPROM HCMOS MCUs WITH ON-SCREEN DISPLAY FOR TV TUNING

PRELIMINARY DATA

- 4.5 to 6V supply operating range
- 8MHz Maximum Clock Frequency
- User Program EPROM: up to 20140 bytes
- Reserved Test EPROM: up to 340 bytes
- Data ROM: user selectable size
- Data RAM: 256 bytes
- Data EEPROM: 384bytes
- 42-Pin Shrink Ceramic Dual in Line Package for EPROM version
- 42-Pin Shrink Plastic Dual in Line Package for OTP version
- Up to 22 software programmable general purpose Inputs/Outputs, including 2 direct LED driving Outputs
- Two Timers each including an 8-bit counter with a 7-bit programmable prescaler
- Digital Watchdog Function
- Serial Peripheral Interface (SPI) supporting S-BUS/ I^2C BUS and standard serial protocols
- SPI for external frequency synthesis tuning
- 14 bit counter for voltage synthesis tuning
- Up to Six 6-Bit PWM D/A Converters
- AFC A/D converter with 0.5V resolution
- Five interrupt vectors (IRIN/NMI, Timer 1 & 2, VSYNC, PWR INT.)
- On-chip clock oscillator
- 5 Lines by 15 Characters On-Screen Display Generator with 128 Characters
- These EPROM and OTP versions are fully pin to pin compatible with their respective ROM version.
- The development tool of the ST636x,7x,8x microcontrollers consists of the ST638x-EMU emulation and development system to be connected via a standard RS232 serial line to an MS-DOS Personal Computer.
- EPROM programming board ST63E8X-EPB



CSDIP42W



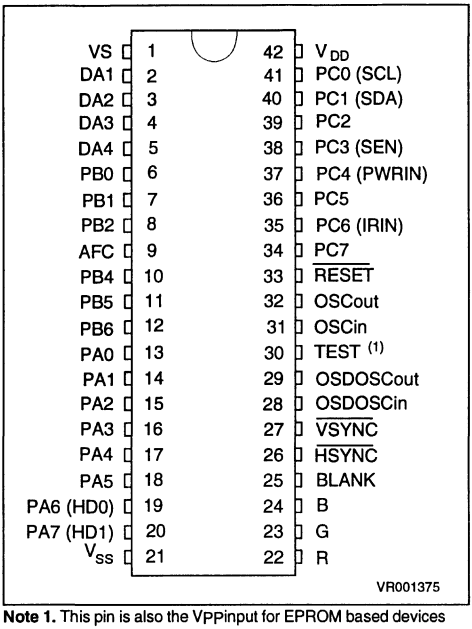
PSDIP42

(Ordering Information at the end of the datasheet)

DEVICE SUMMARY

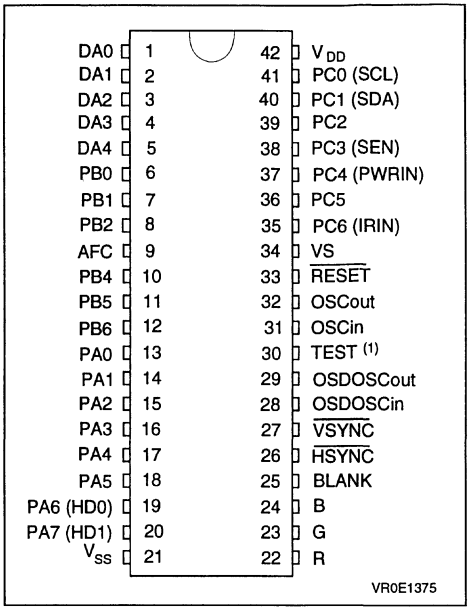
EPROM DEVICE	OTP DEVICE	ROM (Bytes)	D/A Converter
ST63E85	ST63T85	20K	4
ST63E87	ST63T87	20K	6

Figure 1. ST63E85, T85 Pin Configuration



Note 1. This pin is also the VPPinput for EPROM based devices

Figure 2. ST63E87, T87 Pin Configuration



GENERAL DESCRIPTION

The ST63E8x microcontrollers are members of the 8-bit HCMOS ST638x family, a series of devices specially oriented to TV applications. Different ROM size and peripheral configurations are available to give the maximum application and cost flexibility. They are the EPROM/OTP versions of the ST636x, 7x, 8x, ROM devices and are suitable for product prototyping and low volume production. All ST638x members are based on a building block approach: a common core is surrounded by a combination of on-chip peripherals (macrocells) available from a standard library. These peripherals are designed with the same Core technology providing full compatibility and short design time. Many of these macrocells are specially dedicated to TV applications. The macrocells of the ST638x family are: two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), a digital hardware activated watchdog function (DHWD), a 14-bit voltage synthesis tuning peripheral, a Serial Peripheral Interface (SPI), up to six 6-bit PWM D/A converters, an AFC A/D converter with 0.5V resolution, an on-screen display (OSD) with 15 characters per line and 128 characters (in two banks each of 64 characters). In addition the following memory resources are available: program EPROM (up to 20K), data RAM (256 bytes), EEPROM (384 bytes). Refer to pin configurations figures and to ST638x device summary (Table 1) for the definition of ST638x family members and a summary of differences among the different types.

Figure 3. ST63E85, T85, E87, T87 Block Diagram

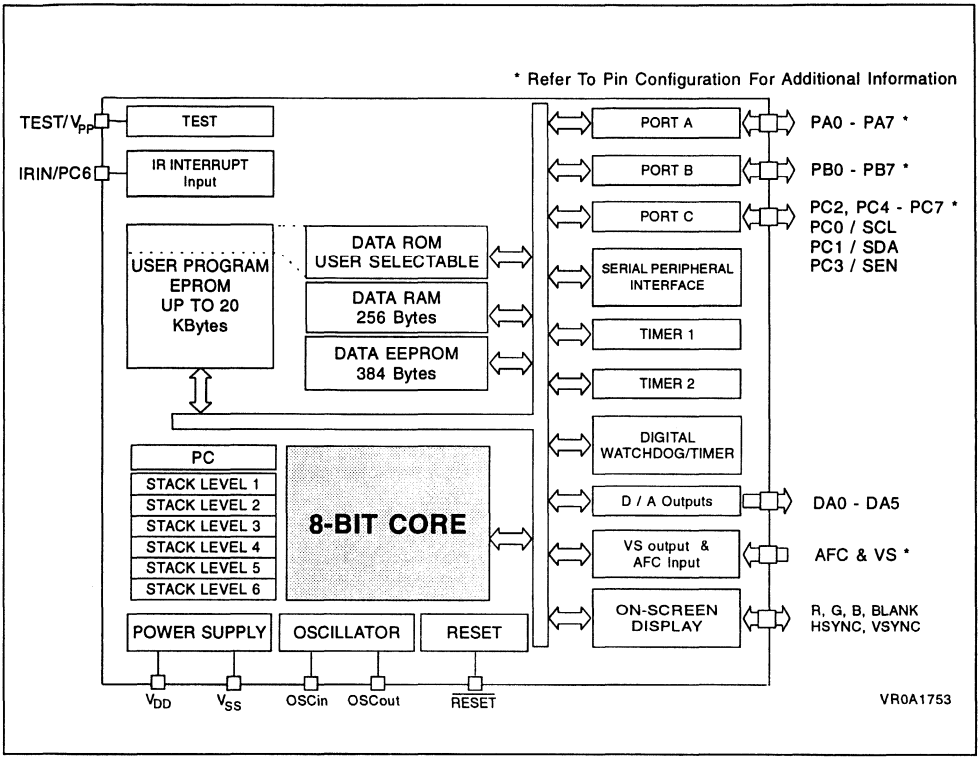


Table 1. Device Summary

DEVICE	EPROM (Bytes)	OTPROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	AFC	VS	D/A	COLOUR PINS	TARGET ROM DEVICES
ST63E85	20K		256	384	YES	YES	4	3	ST6365, 75, 85
ST63T85		20K	256	384	YES	YES	4	3	ST6365, 75, 85
ST63E87	20K		256	384	YES	YES	6	3	ST6367, 77 87
ST63T87		20K	256	384	YES	YES	6	3	ST6367, 77, 87

PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCin, OSCout. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to start the microcontroller to the beginning of its program. Additionally the quartz crystal oscillator will be disabled when the $\overline{\text{RESET}}$ pin is low to reduce power consumption during reset phase.

TEST/V_{PP}. The TEST pin must be held at V_{SS} for normal operation.

If this pin is connected to a +12.5V level during the reset phase, The EPROM programming mode is entered.

CAUTION: Exceeding 13V on TEST/V_{PP} pin will permanently damaged the device

PA0-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured as either an input with or without pull-up resistor or as an output under software control of the data direction register. Pins PA4 to PA7 are configured as open-drain outputs (12V drive). On PA4-PA7 pins the input pull-up option is not available while PA6 and PA7 have additional current driving capability (25mA, V_{OL}:1V). PA0 to PA3 pins are configured as push-pull.

PB0-PB2, PB4-PB6. These 6 lines are organized as one I/O port (B). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register.

PC0-PC7. These 8 lines are organized as one I/O port (C). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register. Pins PC0 to PC3 are configured as open-drain (5V drive) in output mode while PC4 to PC7 are open-drain with 12V drive and the input pull-up options does not exist on these four pins. PC0, PC1 and PC3 lines when in output mode are "ANDed" with the SPI control signals and are all open-drain. PC0 is connected to the SPI clock signal (SCL), PC1 with the SPI data signal (SDA) while PC3 is connected with SPI enable signal (SEN, used in S-BUS protocol). Pin PC4 and PC6 can also be inputs to software programmable edge sensitive latches which can generate interrupts; PC4 can be connected to Power Interrupt while PC6 can be connected to the IRIN/NMI interrupt line.

DA0-DA5. These pins are the six PWM D/A outputs of the 6-bit on-chip D/A converters. These lines have open-drain outputs with 12V drive. The output repetition rate is 31.25KHz (with 8MHz clock).

AFC. This is the input of the on-chip 10 levels comparator that can be used to implement the AFC function. This pin is an high impedance input able to withstand signals with a peak amplitude up to 12V.

OSDOSCin, OSDOSCout. These are the On Screen Display oscillator terminals. An oscillation capacitor and coil network have to be connected to provide the right signal to the OSD.

HSYNC, VSYNC. These are the horizontal and vertical synchronization pins. The active polarity of these pins to the OSD macrocell can be selected by the user as ROM mask option. If the device is specified to have negative logic inputs, then these signals are low the OSD oscillator stops. If the device is specified to have positive logic inputs, then when these signals are high the OSD oscillator stops.

R, G, B, BLANK. Outputs from the OSD. R, G and B are the color outputs while BLANK is the blanking output. All outputs are push-pull. The active polarity of these pins can be selected by the user as ROM mask option.

VS. This is the output pin of the on-chip 14-bit voltage synthesis tuning cell (VS). The tuning signal present at this pin gives an approximate resolution of 40KHz per step over the UHF band. This line is a push-pull output with standard drive.

Table 2. Pin Summary

Pin Function	Description
DA0 to DA5	Output, Open-Drain, 12V
AFC	Input, High Impedance, 12V
VS	Output, Push-Pull
R,G,B, BLANK	Output, Push-Pull
HSYNC, VSYNC	Input, Pull-up, Schmitt Trigger
OSDOSCin	Input, High Impedance
OSDOSCout	Output, Push-Pull
TEST/V_{PP}	Input, Pull-Down, V_{PP} EPROM Programming Voltage Input
OSCin	Input, Resistive Bias, Schmitt Trigger to Reset Logic Only
OSCout	Output, Push-Pull
RESET	Input, Pull-up, Schmitt Trigger Input
PA0-PA3	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PA4-PA5	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
PA6-PA7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input, High Drive
PB0-PB2	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PB4-PB6	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PC0-PC3	I/O, Open-Drain, 5V , Software Input Pull-up, Schmitt Trigger Input
PC4-PC7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
V_{DD}, V_{SS}	Power Supply Pins

MEMORY SPACE

Table 3. Program EPROM Map

EPROM Page	Device Address	EPROM Address	Description
Page 0	0000h-007Fh 0080h-07FFh	0000h-007Fh 0080h-07FFh	Reserved User ROM
Page 1 "STATIC"	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
Page 2	0000h-000Fh 0010h-07FFh	1000h-100Fh 1010h-17FFh	Reserved User ROM
Page 3	0000h-000Fh 0010h-07FFh	1800h-180Fh 1810h-1FFFh	Reserved user ROM
Page 4	0000h-000Fh 0010h-07FFh	2000h-200Fh 2010h-27FFh	Reserved User ROM
Page 5	0000h-000Fh 0010h-07FFh	2800h-280Fh 2810h-2FFFh	Reserved User ROM
Page 6	0000h-000Fh 0010h-07FFh	3000h-300Fh 3010h-37FFh	Reserved User ROM
Page 7	0000h-000Fh 0010h-07FFh	3800h-380Fh 3810h-3FFFh	Reserved User ROM
Page 8	0000h-000Fh 0010h-07FFh	4000h-400Fh 4010h-47FFh	Reserved User ROM
Page 9	0000h-000Fh 0010h-07FFh	4800h-480Fh 4810h-4FFFh	Reserved User ROM (End of 20K)

EPROM/OTP DESCRIPTION.

The ST63E8x is the EPROM version of the ST636x, 7x, 8x ROM products. They are intended for use during the development of an application, and for pre-production and small volume production. The ST63T8x OTP have the same characteristics. They both include EPROM memory instead of the ROM memory of the ST638x, and so the program and constants of the program can be easily modified by the user with the ST63E8x EPROM programming board from SGS-THOMSON.

The ROM mask options of the ST638x for OSD polarities (HSYNC, VSYNC, R, G, B, BLANK) are emulated with an **EPROM OPTION BYTE**. This is programmed by the EPROM programming board and its associated software.

The EPROM Option Byte content will define the OSD options as follows :

7							0
0	0	0	1	0	Opt 2	Opt 1	Opt 0

b7-3: Device specific bits, these reserved bits must be programmed with "00010".

OPT 0 : This bit define the OSD H/Vsync polarity,
if 0 the polarity will be negative
if 1 the polarity will be positive.

OPT 1 : This bit define the RGB polarity,
if 0 the polarity will be negative
if 1 the polarity will be positive..

OPT 2 : This bit define the BLANK polarity,
if 0 the polarity will be negative
if 1 the polarity will be positive..

From a user point of view (with the following exceptions) the ST63E8x,T8x products have exactly the same software and hardware features of the ROM version. An additional mode is used to configure the part for programming of the EPROM, this is set by a +12.5V voltage applied to the TEST/V_{PP} pin. The programming of the ST63E8x,T8x is described in the User Manual of the EPROM Programming board.

On the ST63E8x, all the 20140 bytes of PROGRAM memory are available for the user, as all the EPROM memory can be erased by exposure to UV light. On the ST63T8x (OTP device) a reserved area for test purposes exists, as for the ST638x ROM device. In order to avoid any discrepancy between program functionality when using the EPROM, OTP and ROM it is recommended NOT TO USE THESE RESERVED AREAS, even when using the ST63E8x. The Table 3 is a summary of the EPROM/ROM Map and its reserved area.

THE READER IS ASKED TO REFER TO THE DATASHEET OF THE ST636x, 7x, 8x ROM-BASED DEVICE FOR FURTHER DETAILS.

EPROM ERASING

The EPROM of the windowed package of the ST63E8x may be erased by exposure to Ultra Violet light.

The erasure characteristic of the ST63E8x EPROM is such that erasure begins when the memory is exposed to light with wave lengths shorter than approximately 4000Å. It should be noted that sunlight and some types of fluorescent lamps have wavelengths in the range 3000-4000Å. It is thus recommended that the window of the ST63E8x package be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.
The recommended erasure procedure of the ST63E8x EPROM is exposure to short wave ultraviolet light which has wavelength 2537Å. The integrated dose (i.e. UV intensity x exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm^2 power rating. The ST63E8x should be placed within 2.5cm (1 inch) of the lamp tubes during erasure.

ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from :

$T_j = T_A + P_D \times R_{thJA}$

Where : T_A = Ambient Temperature.

R_{thJA} = Package thermal resistance (junction-to ambient).

P_D = Pint + Pport.

Pint = $I_{DD} \times V_{DD}$ (chip internal power).

Pport = Port power dissipation (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage (AFC IN)	V_{SS} - 0.3 to +13	V
V_I	Input Voltage (Other Inputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
V_O	Output Voltage (PA4-PA7, PC4-PC7, DA0-DA5)	V_{SS} - 0.3 to +13	V
V_O	Output Voltage (Other Outputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
V_{PP}	EPROM programming Voltage	-0.3 to 13.0	V
I_O	Current Drain per Pin Excluding V_{DD}, V_{SS}, PA6, PA7	± 10	mA
I_O	Current Drain per Pin (PA6, PA7)	± 50	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	150	mA
Tj	Junction Temperature	150	°C

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature		0		70	°C
V_{DD}	Operating Supply Voltage		4.5	5.0	6.0	V
V_{PP}	EPROM programming Voltage		12.0	12.5	13.0	V
f_{OSC}	Oscillator Frequency RUN & WAIT Modes			8	8.1	MHz
f_{OSDOSC}	On-screen Display Oscillator Frequency				8.0	MHz

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PSDIP42			67	°C/W

EEPROM INFORMATION

The ST63xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = 0 to +70°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	All I/O Pins			$0.2xV_{DD}$	V
V_{IH}	Input High Level Voltage	All I/O Pins	$0.8xV_{DD}$			V
V_{HYS}	Hysteresis Voltage(1)	All I/O Pins V_{DD} = 5V		1.0		V
V_{OL}	Low Level Output Voltage	DA0-DA5, PB0-PB6, OSD Outputs, PC0-PC7, O0, O1, PA0-PA5 V_{DD} = 4.5V I_{OL} = 1.6mA I_{OL} = 5.0mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	PA6-PA7 V_{DD} = 4.5V I_{OL}= 1.6mA I_{OL}= 25mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	OSDOSCout, OSCout V_{DD} = 4.5V I_{OL}= 0.4mA			0.4	V
V_{OL}	Low Level Output Voltage	VS Output V_{DD} = 4.5V I_{OL}= 0.5mA I_{OL}= 1.6mA			 0.4 1.0	 V V
V_{OH}	High Level Output Voltage	PB0-PB7, PA0-PA3, OSD Outputs V_{DD} = 4.5V I_{OH} = – 1.6mA	4.1			V
V_{OH}	High Level Output Voltage	OSDOSCout, OSCout, V_{DD} = 4.5V I_{OH}= – 0.4mA	4.1			V

DC ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{OH}	High Level Output Voltage	VS Output V_{DD} = 4.5V I_{OH} = –0.5mA	4.1			V
I_{PU}	Input Pull Up Current Input Mode with Pull-up	PB0-PB6, PA0-PA3, PC0-PC3 V_{IN}= V_{SS}	– 100	– 50	– 25	mA
I_{IL} I_{IH}	Input Leakage Current	OSCin V_{IN}= V_{SS} V_{IN}= V_{DD}	– 10 0.1	– 1 1	– 0.1 10	µA
I_{IL}	Input Pull-down current in Reset	OSCin	100			µA
I_{IL} I_{IH}	Input Leakage Current	All I/O Input Mode no Pull-up OSDOSCin V_{IN}= V_{DD} or V_{SS}	– 10		10	µA
V_{DD}RAM	RAM Retention Voltage in RESET		1.5			V
I_{IL} I_{IH}	Input Leakage Current	Reset Pin with Pull-up V_{IN}= V_{SS}	– 50	– 30	– 10	µA
I_{IL} I_{IH}	Input Leakage Current	AFC Pin V_{IH}= V_{DD} V_{IL}= V_{SS} V_{IH}= 12.0V	– 1		1 40	µA
I_{OH}	Output Leakage Current	DA0-DA5, PA4-PA5, PC0-PC7, O0, O1 V_{OH} = V_{DD}			10	µA
I_{OH}	Output Leakage Current High Voltage	DA0-DA5, PA4-PA7, PC4-PC7, O0, O1 V_{OH} = 12V			40	µA
I_{DD}	Supply Current RUN Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6.0V		6	16	mA
I_{DD}	Supply Current WAIT Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6V		3	10	mA
I_{DD}	Supply Current at transition to RESET	f_{OSC}= Not App, ILoad= 0mA V_{DD}= 6V		0.1	1	mA
V_{ON}	Reset Trigger Level ON	RESET Pin			0.3xV_{DD}	V
V_{OFF}	Reset Trigger Level OFF	RESET Pin	0.8xV_{DD}			V
V_{TA}	Input Level Absolute Tolerance	A/D AFC Pin V_{DD} = 5V			±200	mV
V_{TR}	Input Level Relatice Tolerance (1)	A/D AFC Pin Relative to other levels V_{DD} = 5V			±100	mV

Note: 1. Not 100% Tested

AC ELECTRICAL CHARACTERISTICS

(T_A = 0 to +70°C, f_{OSC}=8MHz, V_{DD}=4.5 to 6.0V unless otherwise specified)

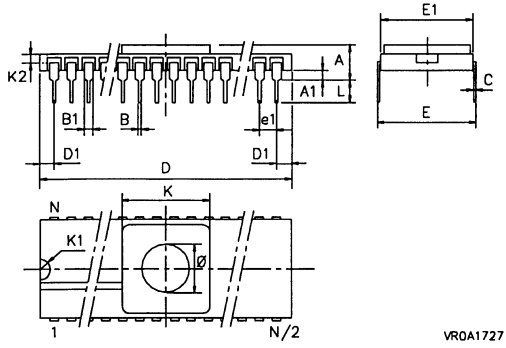
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{WRES}	Minimum Pulse Width	RESET Pin	125			ns
tO_{HL}	High to Low Transition Time	PA6, PA7 V_{DD} = 5V, CL = 1000pF (2)		100		ns
tO_{HL}	High to Low Transition Time	DA0-DA5, PB0-PB6, OSD Outputs, PC0-PC7, V_{DD} = 5V, CL = 100pF		20		ns
tO_{LH}	Low to High Transition Time	PB0-PB6, PA0-PA3, OSD Outputs, PC0-PC3 V_{DD} = 5V, CL = 100pF		20		ns
tO_H	Data HOLD Time SPI after clock goes low I^2CBUS/S-BUS Only		175			ns
f_{DA}	D/A Converter Repetition Frequency(1)			31.25		kHz
f_{SIO}	SIO Baud Rate(1)			62.50		kHz
t_{WEE}	EEPROM Write Time	T_A = 25°C, One Byte		5	10	ms
Endurance	EEPROM WRITE/ERASE Cycles	Q_A LOT Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention (4)	T_A = 25°C	10			years
C_{IN}	Input Capacitance (3)	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance (3)	All outputs Pins			10	pF
COSCin, COSCout	Oscillator Pins Internal Capacitance(3)			5		pF
COSDin, COSDout	OSD Oscillator External Capacitance	Recommended	15		25	pF

Notes:

1. A clock other than 8 MHz will affect the frequency response of those peripherals (D/A, 62.5kHz and SPI) whose clock is derived from the system clock.
2. The rise and fall times of PORT A have been reduced in order to avoid current spikes while maintaining a high drive capability
3. Not 100% Tested
4. Based on extrapolated data

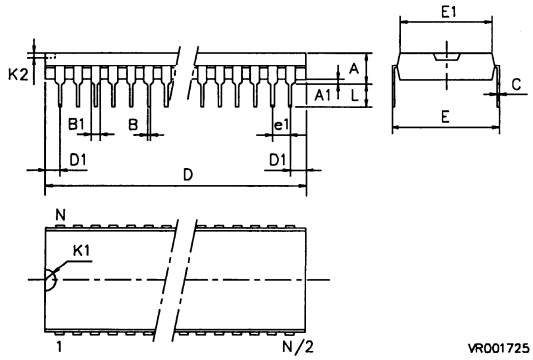
PACKAGE MECHANICAL DATA

Figure 4. 42 Pin ceramic Shrink Dual-In-line Package, 600 Mil width, with Window



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A		4.5			0.177	
A1		1.27			0.050	
B		0.45			0.018	
B1		0.89			0.035	
C		0.25			0.010	
D		37.3			1.470	
D1		—			—	
E		15.49			0.610	
E1		14.98			0.590	
K		—			—	
K1		—			—	
L		3.2			0.125	
Ø		—			—	
e1		1.78			.070	
Number of Pins						
N	42					

Figure 5. 42 Pin Plastic Shrink Dual-In-line Package, 600 Mil width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D	36.32		39.12	1.430		1.540
D1	–	–	–	–	–	–
E			18.54			0.730
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	42					

ORDERING INFORMATION

To ensure compatibility between the EPROM/OTP parts and the corresponding ROM families, the following information is provided. the user should take this information into account when programming the memory and OSD characters of the EPROM parts.

Communication of the ROM Codes. To communicate the contents of memories to SGS-THOMSON, the customer has to send:

- one file in INTEL INTELLEC 8/MDS FORMAT (either as an EPROM or in a MS-DOS 5" diskette) for the ODD and EVEN OSD Character OSD ROM/EEPROM
- one file in INTEL INTELLEC 8/MDS FORMAT (either as an EPROM or in a MS-DOS 5" diskette) for the EEPROM initial content (this file is optional)
- a filled Option List form as described in the OPTION LIST paragraph.

The ROM code must be generated with ST6 assembler. Before programming the EPROM, the buffer of the EPROM programmer must be filled with FFh.

For shipment to SGS-THOMSON the EPROMs should be placed in a conductive IC carrier and packaging carefully.

Customer EEPROM Initial Contents: Format

a. The content should be written into an INTEL INTELLEC format file.

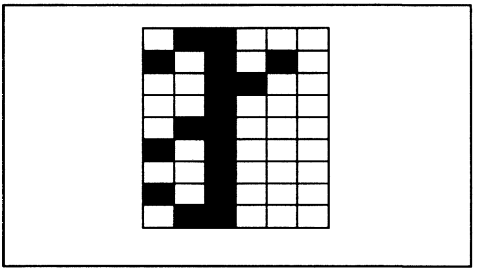
b. In the case of 384 bytes of EEPROM, the starting address is 000h and the end address is 7Fh. The order of the pages (64 bytes each) is an in the specification (ie. b7, b1 b0: 001, 010, 011, 101, 110. 111).

c. Undefined or don't care bytes should have the content FFh.

OSD Test Character. IN ORDER TO ALLOW THE TESTING OF THE ON-CHIP OSD MACROCELL THE FOLLOWING CHARACTER MUST BE PROVIDED AT THE FIXED 3Fh (63) POSITION OF THE SECOND OSD BANK.

Listing Generation & Verification. When SGS-THOMSON receives the Codes, they are compared and a computer listing is generated from them. This listing refers extractly to the mask that will be used to produce the microcontroller. Then the listing is returned to the customer that must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed list constitutes a part of the contractual agreement for the creation of the customer mask. SGS-THOMSON sales organization will provide detailed information on contractual points.

Figure 6. OSD Test Character



ST63E8x, T8x MICROCONTROLLER OPTION LIST

Customer: . . . . . .
Address: . . . . . .
Contact: . . . . . .
Phone No: . . . . . .
Reference: . . . . . .

Device [ ] (d) Package [ ] (p) Temperature Range [ ] (t)
For marking one line with 14 characters maximum is possible
Special Marking [ ] (y/n) Line1 " _ _ _ _ _ _ _ _ _ _ _ _ _ _ " (N)
(For Plastic Package only)

Notes:
(d) 1= ST63E85, 2 = ST63E87, 3 = ST63T85, 4 = ST63T87
(p) B= Plastic Dual in Line, D= Ceramic Dual in line with Window
(t) 1= 0 to 70°C
(N) Letters, digits, ' . ', ' - ', ' / ' and spaces only

Marking: the default marking is equivalent to the sales type only (part number).

CHECK LIST:

	YES	NO
OSD Code: ODD & EVEN	[ ]	[ ]
EEPROM Code (if Desired)	[ ]	[ ]

Signature .....

Date .....

ORDERING INFORMATION TABLE

Sales Type	EPROM/EEPROM Size	D/A Converter	Temperature Range	Package
ST63E85D1/XX	20K/384 Bytes	4	0 to + 70 ° C	CSDIP42W
ST63E87D1/XX	20K/384 Bytes	6	0 to + 70 ° C	CSDIP42W
ST63T85B1/XX	20K/384 Bytes	4	0 to + 70 ° C	PSDIP42
ST63T87B1/XX	20K/384 Bytes	6	0 to + 70 ° C	PSDIP42

Note: "XX" Is the ROM code identifier that is allocated by SGS-THOMSON after receipt of all required options and the related ROM file.

Pages

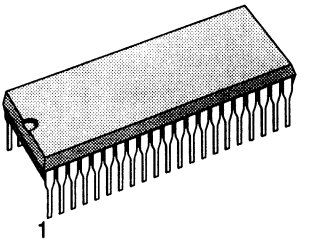
SGS-THOMSON MICROELECTRONICS

ST6369

8-BIT HCMOS MCU FOR DIGITAL CONTROLLED MULTI FREQUENCY MONITOR

PRELIMINARY DATA

- 4.5 to 6V supply operating range
- 8MHz Maximum Clock Frequency
- User Program ROM: 7948 bytes
- Reserved Test ROM: 244 bytes
- Data ROM: user selectable size
- Data RAM: 256 bytes
- Data EEPROM: 384 bytes
- 40-Pin Dual in Line Plastic Package
- Up to 23 software programmable general purpose Inputs/Outputs, including 2 direct LED driving Outputs
- Two Timers each including an 8-bit counter with a 7-bit programmable prescaler
- Digital Watchdog Function
- Serial Peripheral Interface (SPI) supporting S-BUS/ I^2C BUS and standard serial protocols
- One 14-Bit PWM D/A Converter
- Six 6-Bit PWM D/A Converters
- One A/D converter with 0.5V resolution
- Five interrupt vectors (HSYNC/NMI, Timer 1 & 2, VSYNC, PWR INT.)
- On-chip clock oscillator
- ST6369 is supported by pin-to-pin EPROM and OTP versions.
- The development tool of the ST6369 microcontroller consists of the ST6369-EMU emulation and development system to be connected via a standard RS232 serial line to an MS-DOS Personal Computer.



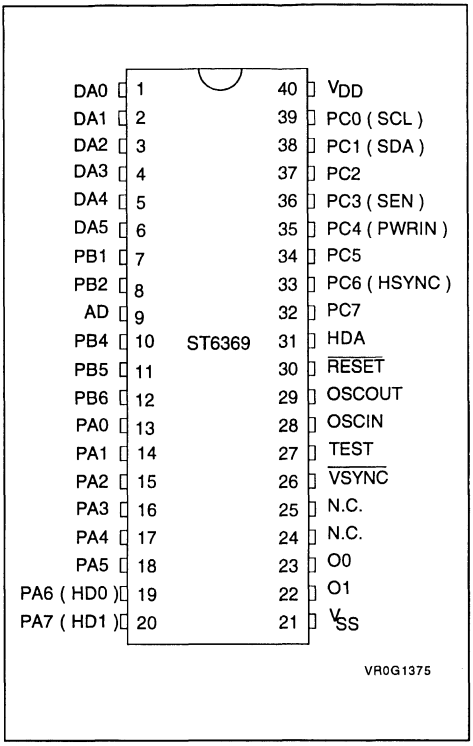
PDIP40

(Ordering Information at the end of the datasheet)

DEVICE SUMMARY

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	D/A Conv.
ST6369	8K	256	384	7

Figure 1. ST6369 Pin Configuration



GENERAL DESCRIPTION

The ST6369 microcontroller is member of the 8-bit HCMOS ST638x family, a series of devices specially oriented to Digital Controlled Multi Frequency Monitor applications. ST6369 members are based on a building block approach: a common core is surrounded by a combination of on-chip peripherals (macrocells) available from a standard library. These peripherals are designed with the same Core technology providing full compatibility and short design time. Many of these macrocells are specially dedicated to DCMF Monitor applications. The macrocells of the ST6369 are: two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), a digital hardware activated watchdog function (DHWD), a 14-bit voltage synthesis tuning peripheral, a Serial Peripheral Interface (SPI), six 6-bit PWM D/A converters, an A/D converter with 0.5V resolution, a 14-bit PWM D/A converter. In addition the following memory resources are available: program ROM (8K bytes), data RAM (256 bytes), EEPROM (384 bytes).

Figure 2. ST6369 Block Diagram

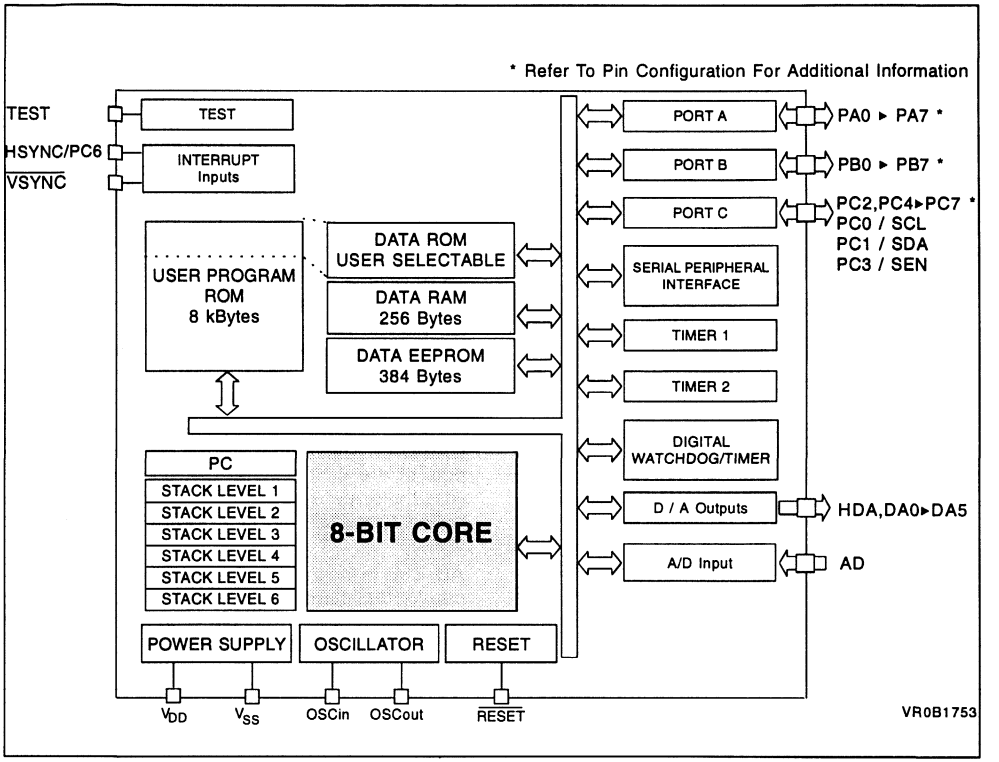


Table 1. Device Summary

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	A/D	14-bit D/A	6-bit D/A	EMULATING DEVICES
ST6369	8K	256	384	1	1	6	ST63E69, ST63T69

PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCIN, OSCOUT. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. The OSCIN pin is the input pin, the OSCOUT pin is the output pin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to start the microcontroller to the beginning of its program. Additionally the quartz crystal oscillator will be disabled when the $\overline{\text{RESET}}$ pin is low to reduce power consumption during reset phase.

TEST. The TEST pin must be held at V_{SS} for normal operation.

PA0-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured as either an input with or without pull-up resistor or as an output under software control of the data direction register. Pins PA4 to PA7 are configured as open-drain outputs (12V drive). On PA4-PA7 pins the input pull-up option is not available while PA6 and PA7 have additional current driving capability (25mA, V_{OL}:1V). PA0 to PA3 pins are configured as push-pull.

PB1-PB2, PB4-PB6. These 5 lines are organized as one I/O port (B). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register.

PC0-PC7. These 8 lines are organized as one I/O port (C). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register. Pins PC0 to PC3 are configured as open-drain (5V drive) in output mode while PC4 to PC7 are open-drain with 12V drive and the input pull-up options does not exist on these four pins. PC0, PC1 and PC3 lines when in output mode are "ANDed" with the SPI control signals and are all open-drain. PC0 is connected to the SPI clock signal (SCL), PC1 with the SPI data signal (SDA) while PC3 is connected with SPI enable signal (SEN, used in S-BUS protocol). Pin PC4 and PC6 can also be inputs to software programmable edge sensitive latches which can generate interrupts; PC4 can be connected to Power Interrupt while PC6 can be connected to the HSYNC/NMI interrupt line.

DA0-DA5. These pins are the six PWM D/A outputs of the 6-bit on-chip D/A converters. These lines have open-drain outputs with 12V drive. The output repetition rate is 31.25KHz (with 8MHz clock).

AD. This is the input of the on-chip 10 levels comparator that can be used to implement the Analog Keyboard function. This pin is an high impedance input able to withstand signals with a peak amplitude up to 12V.

VSYNC. This is the Vertical Synchronization pin. This pin is connected to an internal timer interrupt.

O0, O1. These two lines are output open-drain pins with 12V drive.

HDA. This is the output pin of the on-chip 14-bit PWM D/A Converter. This line is a push-pull output with standard drive.

Table 2. Pin Summary

Pin Function	Description
DA0 to DA5	Output, Open-Drain, 12V
AD	Input, High Impedance, 12V
HDA	Output, Push-Pull
VSYNC	Input, Pull-up, Schmitt Trigger
TEST	Input, Pull-Down
OSCIN	Input, Resistive Bias, Schmitt Trigger to Reset Logic Only
OSCOUT	Output, Push-Pull
RESET	Input, Pull-up, Schmitt Trigger Input
PA0-PA3	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PA4-PA5	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
PA6-PA7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input, High Drive
PB1-PB2	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PB4-PB6	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PC0-PC3	I/O, Open-Drain, 5V , Software Input Pull-up, Schmitt Trigger Input
PC4-PC7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
O0, O1	Output, Open-Drain, 12V
V_{DD}, V_{SS}	Power Supply Pins

ST6369 CORE

The Core of the ST6369 is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal addresses, data, and control busses. The in-core communication is arranged as shown in the following block diagram figure; the controller being externally linked to both the reset and the oscillator, while the core is linked to the dedicated on-chip macrocells peripherals via the serial data bus and indirectly for interrupt purposes through the control registers.

Registers

The ST6369 Core has five registers and three pairs of flags available to the programmer. They are shown in Figure 4 and are explained in the following paragraphs together with the program and data memory page registers.

Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator is addressed in the data space as RAM location at the FFH address. Accordingly, the ST6369 instruction set can use the accumulator as any other register of the data space.

Figure 4. Core Programming Model

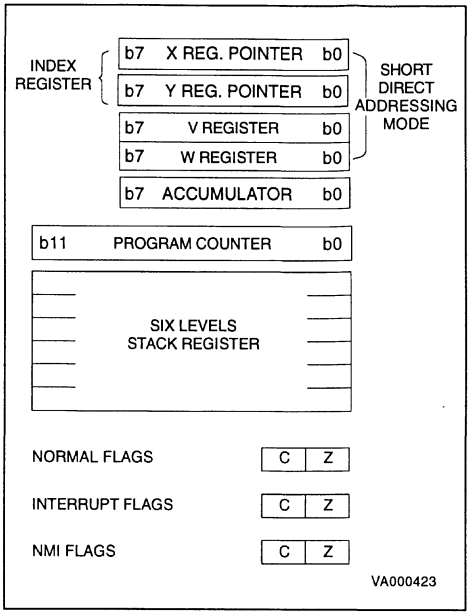
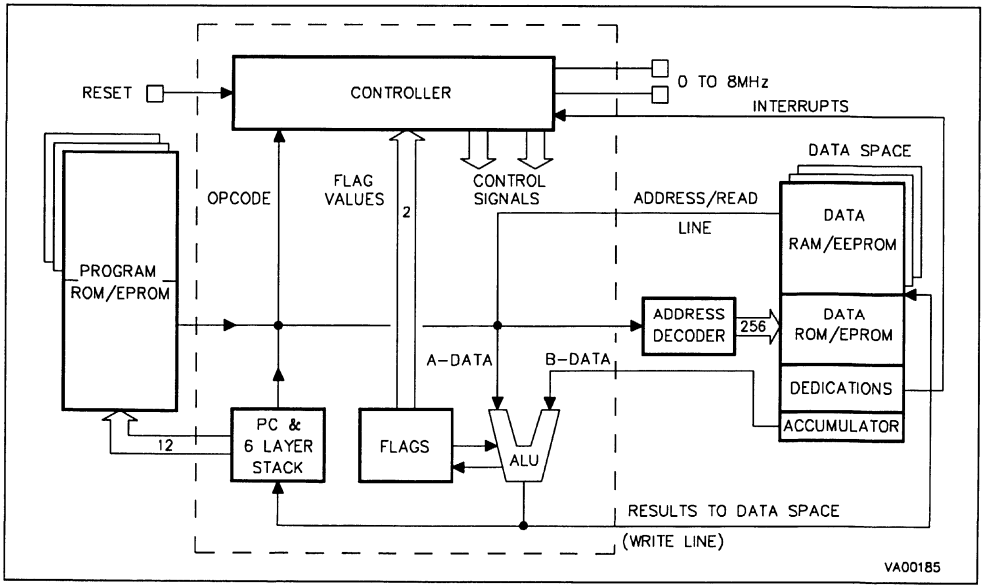


Figure 3. Core Block Diagram



ST6369 CORE (Continued)

Indirect Registers (X, Y). These two indirect registers are used as pointers to the memory locations in the data space. They are used in the register-indirect addressing mode.These registers can be addressed in the data space as RAM locations at the 80H (X) and 81H (Y) addresses. They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST638x instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save one byte in short direct addressing mode. These registers can be addressed in the data space as RAM locations at the 82H (V) and 83H (W) addresses. They can also be accessed with the direct and bit direct addressing modes. Accordingly, the ST638x instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC)

The program counter is a 12-bit register that contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or an address of operand. The 12-bit length allows the direct addressing of 4096 bytes in the program space. Nevertheless, if the program space contains more than 4096 locations, the further program space can be addressed by using the Program ROM Page Register. The PC value is incremented, after it is read for the address of the current instruction, by sending it through the ALU, so giving the address of the next byte in the program. To execute relative jumps the PC and the offset values are shifted through the ALU, where they will be added, and the result is shifted back into the PC. The program counter can be changed in the following ways:

JP (Jump) instruction.... PC= Jump address

CALL instruction..... PC= Call address

Relative Branch instructions..... PC= PC+offset

Interrupt..... PC= Interrupt vector

Reset..... PC= Reset vector

RET & RETI instructions..... PC= Pop (stack)

Normal instruction ..... PC= PC+1

Flags (C, Z)

The ST6369 Core includes three pairs of flags that correspond to 3 different modes: normal mode, interrupt mode and Non-Maskable-Interrupt-Mode. Each pair consists of a CARRY flag and a ZERO flag. One pair (CN, ZN) is used during normal operation, one pair is used during the interrupt mode (CI,ZI) and one is used during the not-maskable interrupt mode (CNMI, ZNMI).

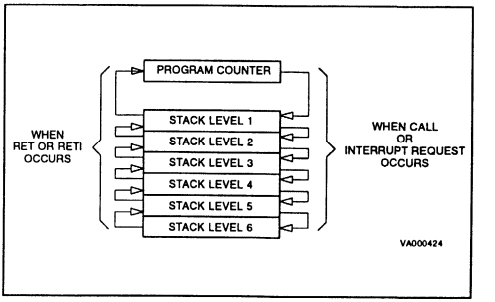
The ST6369 Core uses the pair of flags that corresponds to the actual mode: as soon as an interrupt (resp. a Non-Maskable-Interrupt) is generated, the ST6369 Core uses the interrupt flags (resp. the NMI flags) instead of the normal flags. When the RETI instruction is executed, the normal flags (resp. the interrupt flags) are restored if the MCU was in the normal mode (resp. in the interrupt mode) before the interrupt. Should be observed that each flag set can only be addressed in its own routine (Not-maskable interrupt, normal interrupt or main routine). The interrupt flags are not cleared during the context switching and so, they remain in the state they were at the exit of the last routine switching.

The Carry flag is set when a carry or a borrow occurs during arithmetic operations, otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction, and participates in the rotate left instruction.

The Zero flag is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

The switching between these three sets is automatically performed when an NMI, an interrupt and a RETI instructions occur. As the NMI mode is automatically selected after the reset of the MCU, the ST6369 Core uses at first the NMI flags.

Figure 5. Stack Operation



ST6369 CORE (Continued)

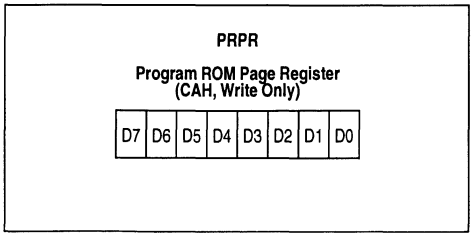
Stack

The ST6369 Core includes true LIFO hardware stack that eliminates the need for a stack pointer. The stack consists of six separate 12-bit RAM locations that do not belong to the data space RAM area. When a subroutine call (or interrupt request) occurs, the contents of each level is shifted into the next level while the content of the PC is shifted into the first level (the value of the sixth level will be lost). When subroutine or interrupt return occurs (RET or RETI instructions), the first level register is shifted back into the PC and the value of each level is shifted back into the previous level. These two operating modes are described in Figure 5. Since the accumulator, as all other data space registers, is not stored in this stack the handling of this registers shall be performed inside the subroutine. The stack pointer will remain in its deepest position, if more than 6 calls or interrupts are executed, so that the last return address will be lost. It will also remain in its highest position if the stack is empty and a RET or RETI is executed. In this case the next instruction will be executed.

Memory Registers

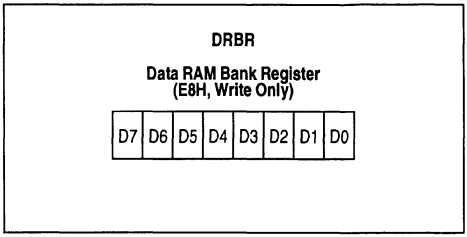
The PRPR can be addressed like a RAM location in the Data Space at the CAH address; nevertheless it is a write-only register that can not be accessed with single-bit operations. This register is used to select the 2-Kbyte ROM bank of the Program Space that will be addressed. The number of the page has to be loaded in the PRPR. The PRPR is not cleared during the MCU initialization and should therefore be defined before jumping out of the static page. Refer to the Program Space description for additional information concerning the use of this register. The PRPR is not modified when an interrupt or a subroutine occurs.

Figure 6. Program ROM Page Register



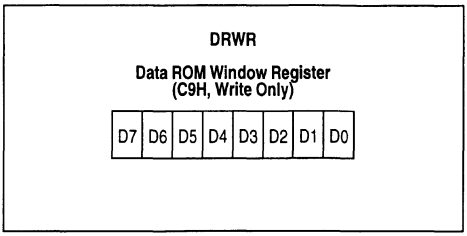
The DRBR can be addressed like a RAM location in the Data Space at the E8H address, nevertheless it is write-only register that can not be accessed with single-bit operations. This register is used to select the desired 64-byte RAM/EEPROM bank of the Data Space. The number of the bank has to be loaded in the DRBR and the instruction has to point to the selected location as it was in the 0 bank (from 00H address to 3FH address). This register is undefined after Reset. Refer to the Data Space description for additional information. The DRBR register is not modified when a interrupt or a subroutine occurs.

Figure 7. Data RAM Bank Register



The DRWR register can be addressed like a RAM location in the Data Space at the C9H address, nevertheless it is write-only register that can not be accessed with single-bit operations. This register is used to move up and down the 64-byte read-only data window (from the 40H address to 7FH address of the Data Space) along the ROM of the MCU by step of 64 bytes. The effective address of the byte to be read as a data in the ROM is obtained by the concatenation of the 6 less significant bits of the address given in the instruction (as less significant bits) and the content of the DRWR (as most significant bits). Refer to the Data Space description for additional information.

Figure 8. Data ROM Window Register



MEMORY SPACES

The MCUs operate in three different memory spaces: Stack Space, Program Space and Data Space. A description of these spaces is shown in Figure 9.

Stack Space

The stack space consists of six 12 bit registers that are used for stacking subroutine and interrupt return addresses plus the current program counter register.

Program Space

The program space is physically implemented in the ROM and includes all the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, the reserved test area and user vectors. It is addressed thanks to the 12-bit Program Counter register (PC register) and so, the ST6369 Core can directly address up to 4K bytes of Program Space. Nevertheless, the Program Space can be extended by the addition of 2-Kbyte ROM banks as it is shown in Figure 11 in which a 8K bytes memory is described. These banks are addressed by pointing to the 000H-7FFH locations of the Program Space thanks to the Program Counter, and by writing the appropriate code in the Program ROM Page Register (PRPR) located at the CAH address of the Data Space. Because interrupts and common subroutines should be available all the time only the lower 2K byte of the 4K program space are bank switched while the upper 2K byte can be seen as static space. Table 3 gives the different codes that allows the selection of the corresponding banks. Note that, from the memory point of view, the Page 1 and the Static Page represent the same physical memory: it is only a different way of addressing the same location.

Figure 10. 8K Bytes Program Space Addressing Description

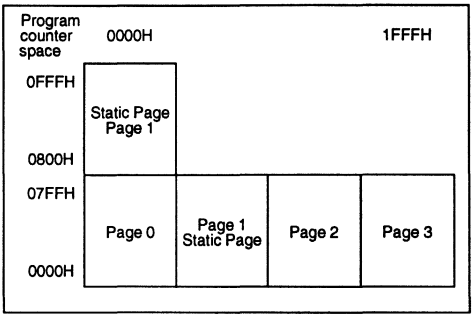
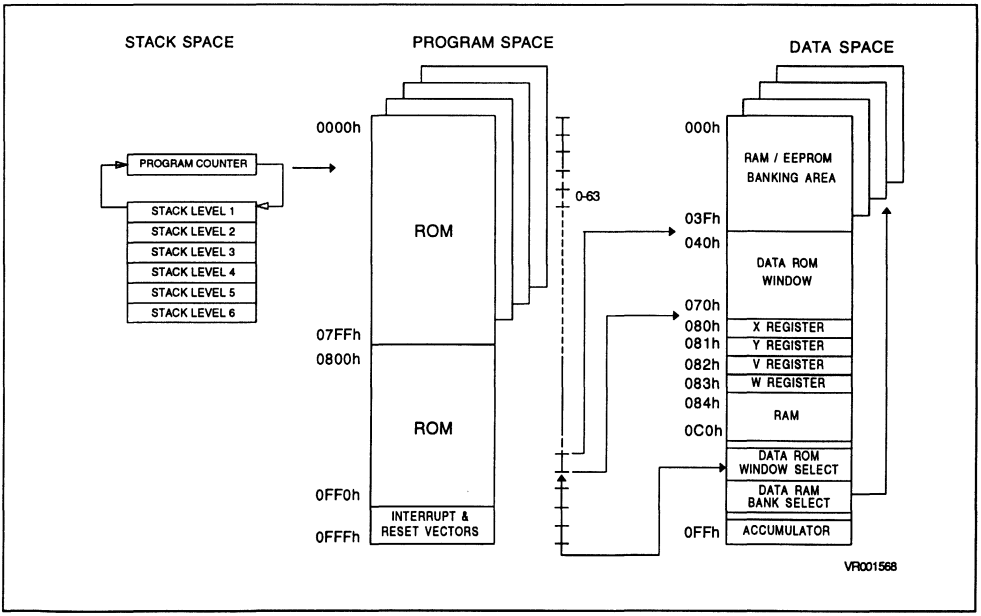
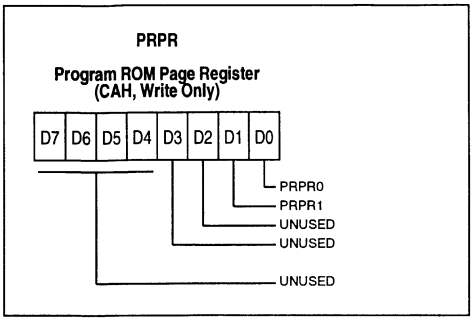


Figure 9. Memory Addressing Description Diagram



MEMORY SPACES (Continued)

Figure 11. Program ROM Page Register



D7-D2. These bits are not used but have to be written to "0".

PRPR1-PRPR0. These are the program ROM banking bits and the value loaded selects the corresponding page to be addressed in the lower part of 4K program address space as specified in Table 3. This register is undefined on reset.

Table 3. Program ROM Page Register Coding

PRPR1	PRPR0	PC11	Memory Page
X	X	1	Static Page (Page 1)
0	0	0	Page 0
0	1	0	Page 1 (Static Page)
1	0	0	Page 2
1	1	0	Page 3

Note. Only the lower part of address space has been bankswitched because interrupt vectors and common subroutines should be available all the time. The reason of this structure is due to the fact that it is not possible to jump from a dynamic page to another, unless jumping back to the static page, changing contents of PRPR, and, then, jumping to a different dynamic page.

Care is required when handling the PRPR as it is write only. For this reason, it is not allowed to change the PRPR contents while executing interrupts drivers, as the driver cannot save and than restore its previous content. Anyway, this operation may be necessary if the sum of common routines and interrupt drivers will take more than 2K bytes; in this case could be necessary to divide the interrupt driver in a (minor) part in the static page (start and end), and in the second (major) part in one dynamic page. If it is impossible to avoid the writing of this register in interrupts drivers, an image of this register must be saved in a RAM location, and each time the program writes the PRPR it writes also the image register. The image register must be written first, so if an interrupt occurs between the two instructions the PRPR is not affected.

Table 4. ST6369 Program ROM Map

ROM Page	Device Address	Description
PAGE 0	0000H-007FH 0080H-07FFH	Reserved User ROM
PAGE 1 "STATIC"	0800H-0F9FH 0FA0H-0FEFH 0FF0H-0FF7H 0FF8H-0FFBH 0FFCH-0FFDH 0FFEH-0FFFH	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
PAGE 2	0000H-000FH 0010H-07FFH	Reserved User ROM
PAGE 3	0000H-000FH 0010H-07FFH	Reserved User ROM

MEMORY SPACES (Continued)

Data Space

The instruction set of the ST6369 Core operates on a specific space, named Data Space that contains all the data necessary for the processing of the program. The Data Space allows the addressing of RAM (256 bytes), EEPROM (384 bytes), ST6369 Core/peripheral registers, and read-only data such as constants and the look-up tables.

Figure 12. Data Space

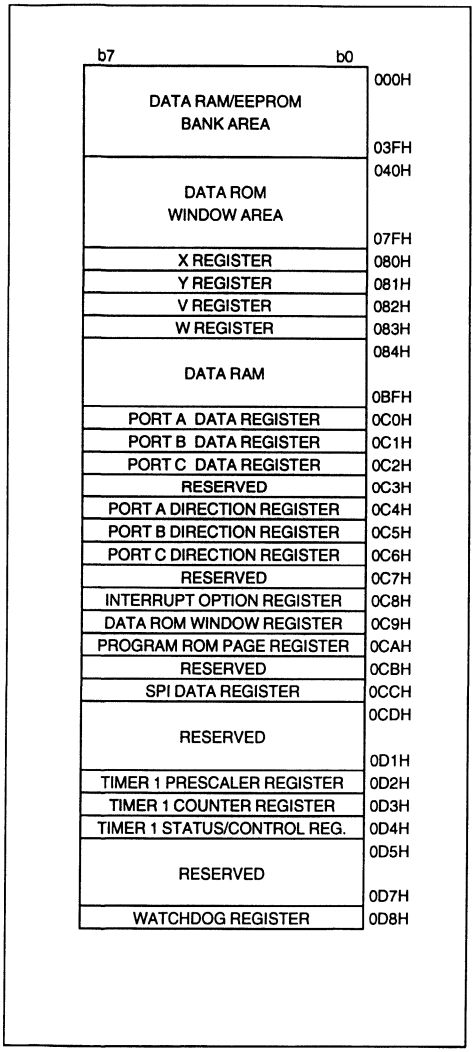
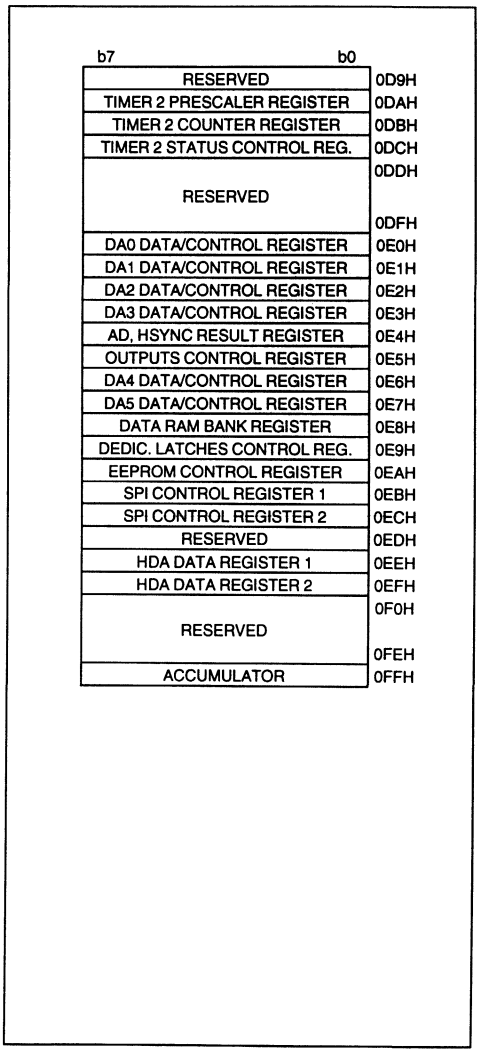


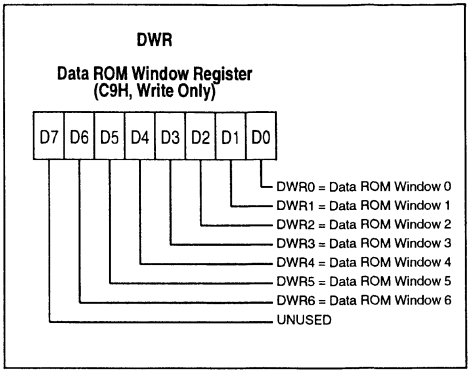
Figure 13. Data Space (Continued)



MEMORY SPACES (Continued)

Data ROM Addressing. All the read-only data are physically implemented in the ROM in which the Program Space is also implemented. The ROM therefore contains the program to be executed and also the constants and the look-up tables needed for the program. The locations of Data Space in which the different constants and look-up tables are addressed by the ST6369 Core can be considered as being a 64-byte window through which it is possible to access to the read-only data stored in the ROM. This window is located from the 40H address to the 7FH address in the Data space and allows the direct reading of the bytes from the 000H address to the 03FH address in the ROM. All the bytes of the ROM can be used to store either instructions or read-only data. Indeed, the window can be moved by step of 64 bytes along the ROM in writing the appropriate code in the Write-only Data ROM Window register (DRWR, location C9H). The effective address of the byte to be read as a data in the ROM is obtained by the concatenation of the 6 less significant bits of the address in the Data Space (as less significant bits) and the content of the DRWR (as most significant bits). So when addressing location 40H of data space, and 0 is loaded in the DRWR, the physical addressed location in ROM is 00H.

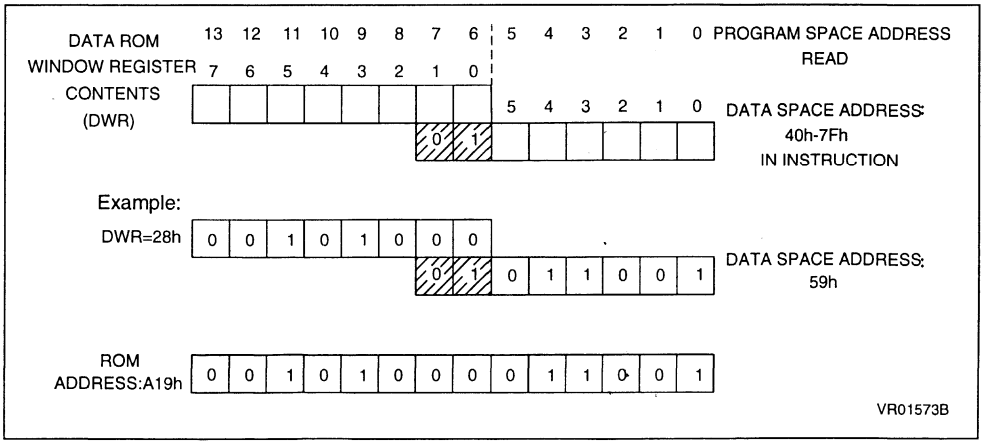
Figure 14. Data ROM Window Register



DWR6-DWR0. These are the Data Rom Window bits that correspond to the upper bits of data ROM program space. This register is undefined after reset.

Note. Care is required when handling the DRWR as it is write only. For this reason, it is not allowed to change the DRWR contents while executing interrupts drivers, as the driver cannot save and than restore its previous content. If it is impossible to avoid the writing of this register in interrupts drivers, an image of this register must be saved in a RAM location, and each time the program writes the DRWR it writes also the image register. The image register must be written first, so if an interrupt occurs between the two instructions the DRWR register is not affected.

Figure 15. Data ROM Window Memory Addressing

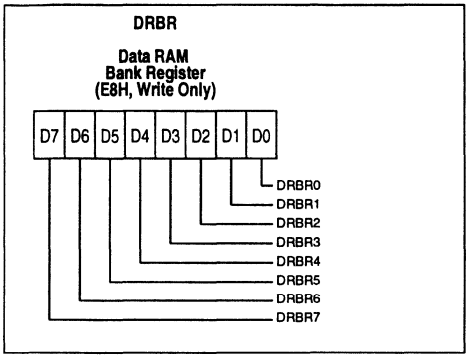


MEMORY SPACES (Continued)

Data RAM/EEPROM

In the ST6369 64 bytes of data RAM are directly addressable in the data space from 80H to BFH addresses. The additional 192 bytes of RAM, the 384 bytes of EEPROM can be addressed using the banks of 64 bytes located between addresses 00H and 3FH. The selection of the bank is done by programming the Data RAM Bank Register (DRBR) located at the E8H address of the Data Space. In this way each bank of RAM, EEPROM can select 64 bytes at a time. No more than one bank should be set at a time.

Figure 16. Data RAM Bank Register



DRBR7,DRBR1,DRBR0. These bits select the EEPROM pages.

DRBR4,DRBR3,DRBR2. Each of these bits, when set, will select one RAM page.

This register is undefined after reset.

Table 5 summarizes how to set the Data RAM Bank Register in order to select the various banks or pages.

Note :
Care is required when handling the DRBR as it is write only. For this reason, it is not allowed to change the DRBR contents while executing interrupts drivers, as the driver cannot save and than restore its previous content. If it is impossible to avoid the writing of this register in interrupts drivers, an image of this register must be saved in a RAM location, and each time the program writes the DRBR it writes also the image register.
The image register must be written first, so if an interrupt occurs between the two instructions the DRBR is not affected.

EEPROM Description

The data space of ST6369 family from 00H to 3FH is paged as described in Table 5. 384 bytes of EEPROM located in six pages of 64 bytes (pages 0,1,2,3,4 and 5, see Table 5).

Table 5. Data RAM Bank Register Set-up

DRBR Value		Selection
Hex.	Binary	
01H	0000 0001	EEPROM Page 0
02H	0000 0010	EEPROM Page 1
03H	0000 0011	EEPROM Page 2
81H	1000 0001	EEPROM Page 3
82H	1000 0010	EEPROM Page 4
83H	1000 0011	EEPROM Page 5
04H	0000 0100	RAM Page 2
08H	0000 1000	RAM Page 3
10H	0001 0000	RAM Page 4

MEMORY SPACES (Continued)

Through the programming of the Data RAM Bank Register (DRBR=E8H) the user can select the bank or page leaving unaffected the way to address the static registers. The way to address the "dynamic" page is to set the DRBR as described in Table 5 (e.g. to select EEPROM page 0, the DRBR has to be loaded with content 01H, see Data RAM/EEPROM addressing for additional information). Bits 0, 1 and 7 of the DRBR are dedicated to the EEPROM.

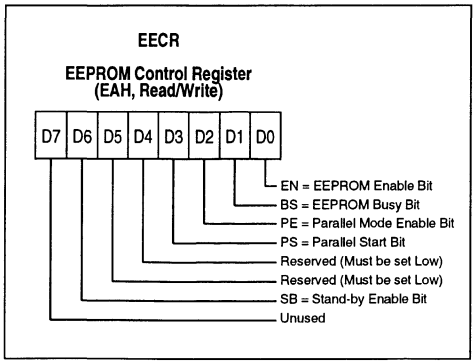
The EEPROM pages do not require dedicated instructions to be accessed in reading or writing. The EEPROM is controlled by the EEPROM Control Register (EECR=EAH). Any EEPROM location can be read just like any other data location, also in terms of access time.

To write an EEPROM location takes an average time of 5 ms (10ms max) and during this time the EEPROM is not accessible by the Core. A busy flag can be read by the Core to know the EEPROM status before trying any access. In writing the EEPROM can work in two modes: Byte Mode (BMODE) and Parallel Mode (PMODE). The BMODE is the normal way to use the EEPROM and consists in accessing one byte at a time. The PMODE consists in accessing 8 bytes per time.

D7. Not used

SB. *WRITE ONLY.* If this bit is set the EEPROM is disabled (any access will be meaningless) and the power consumption of the EEPROM is reduced to the leakage values.

Figure 17. EEPROM Control Register



D5, D4. Reserved for testing purposes, they must be set to zero.

PS. *SET ONLY.* Once in Parallel Mode, as soon as the user software sets the PS bit the parallel writing of the 8 adjacent registers will start. PS is internally reset at the end of the programming procedure. Note that less than 8 bytes can be written; after parallel programming the remaining undefined bytes will have no particular content.

PE. *WRITE ONLY.* This bit must be set by the user program in order to perform parallel programming (more bytes per time). If PE is set and the "parallel start bit" (PS) is low, up to 8 adjacent bytes can be written at the maximum speed, the content being stored in volatile registers. These 8 adjacent bytes can be considered as row, whose A7, A6, A5, A4, A3 are fixed while A2, A1 and A0 are the changing bytes. PE is automatically reset at the end of any parallel programming procedure. PE can be reset by the user software before starting the programming procedure, leaving unchanged the EEPROM registers.

BS. *READ ONLY.* This bit will be automatically set by the CORE when the user program modifies an EEPROM register. The user program has to test it before any read or write EEPROM operation; any attempt to access the EEPROM while "busy bit" is set will be aborted and the writing procedure in progress completed.

EN. *WRITE ONLY.* This bit MUST be set to one in order to write any EEPROM register. If the user program will attempt to write the EEPROM when EN= "0" the involved registers will be unaffected and the "busy bit" will not be set.

After RESET the content of EECR register will be 00H.

Notes :
When the EEPROM is busy (BS="1") the EECR can not be accessed in write mode, it is only possible to read BS status. This implies that as long as the EEPROM is busy it is not possible to change the status of the EEPROM control register. EECR bits 4 and 5 are reserved for test purposes, and must never be set to "1".

Additional Notes on Parallel Mode. If the user wants to perform a parallel programming the first action should be the set to one the PE bit; from this moment the first time the EEPROM will be addressed in writing, the ROW address will be latched and it will be possible to change it only at the end of the programming procedure or by reset-

MEMORY SPACES (Continued)

ting PE without programming the EEPROM. After the ROW address latching the Core can "see" just one EEPROM row (the selected one) and any attempt to write or read other rows will produce errors. Do not read the EEPROM while PE is set.

As soon as PE bit is set, the 8 volatile ROW latches are cleared. From this moment the user can load data in the whole ROW or just in a subset. PS setting will modify the EEPROM registers corresponding to the ROW latches accessed after PE. For example, if the software sets PE and accesses EEPROM in writing at addresses 18H,1AH,1BH and then sets PS, these three registers will be modified at the same time; the remaining bytes will have no particular content. Note that PE is internally reset at the end of the programming procedure. This implies that the user must set PE bit between two parallel programming procedures. Anyway the user can set and then reset PE without performing any EEPROM programming. PS is a set only bit and is internally reset at the end of the programming procedure. Note that if the user tries to set PS while PE is not set there will not be any programming procedure and the PS bit will be unaffected. Consequently PS bit can not be set if EN is low. PS can be affected by the user set if, and only if, EN and PE bits are also set to one.

INTERRUPT

The ST6369 Core can manage 4 different maskable interrupt sources, plus one non-maskable interrupt source (top priority level interrupt). Each source is associated with a particular interrupt vector that contains a Jump instruction to the related interrupt service routine. Each vector is located in the Program Space at a particular address (see Table 6). When a source provides an interrupt request, and the request processing is also enabled by the ST6369 Core, then the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction). Finally, the PC is loaded with the address of the Jump instruction and the interrupt routine is processed.

The relationship between vector and source and the associated priority is hardware fixed for the different ST638x devices. For some interrupt sources it is also possible to select by software the kind of event that will generate the interrupt.

All interrupts can be disabled by writing to the GEN bit (global interrupt enable) of the interrupt option register (address C8H). After a reset, ST6369 is in non maskable interrupt mode; so no interrupts will be accepted and NMI flags will be used, until a RETI instruction is executed. If an interrupt is executed, one special cycle is made by the core, during that the PC is set to the related interrupt vector address. A jump instruction at this address has to redirect program execution to the beginning of the related interrupt routine. The interrupt detecting cycle, also resets the related interrupt flag (not available to the user), so that another interrupt can be stored for this current vector, while its driver is under execution.

If additional interrupts arrive from the same source, they will be lost. NMI can interrupt other interrupt routines at any time, while other interrupts cannot interrupt each other. If more than one interrupt is waiting for service, they are executed according to their priority. The lower the number, the higher the priority. Priority is, therefore, fixed. Interrupts are checked during the last cycle of an instruction (RETI included). Level sensitive interrupts have to be valid during this period.

Table 6 details the different interrupt vectors/sources relationships.

Interrupt Vectors/Sources

The ST6369 Core includes 5 different interrupt vectors in order to branch to 5 different interrupt routines. The interrupt vectors are located in the fixed (or static) page of the Program Space.

INTERRUPT (Continued)

Table 6. Interrupt Vectors/Sources Relationships

Interrupt Source	Associated Vector	Vector Address
PC6/IRIN Pin (1)	Interrupt Vector # 0 (NMI)	0FFCH-0FFDH
Timer 2	Interrupt Vector # 1	0FF6H-0FF7H
Vsync	Interrupt Vector # 2	0FF4H-0FF5H
Timer 1	Interrupt Vector # 3	0FF2H-0FF3H
PC4/PWRIN	Interrupt Vector # 4	0FF0H-0FF1H

Note: 1. This pin is associated with the NMI Interrupt Vector

The interrupt vector associated with the non-maskable interrupt source is named interrupt vector #0. It is located at the (FFCH,FFDH) addresses in the Program Space. This vector is associated with the PC6/IRIN pin.

The interrupt vectors located at addresses (FF6H,FF7H), (FF4H,FF5H), (FF2H,FF3H), (FF0H,FF1H) are named interrupt vectors #1, #2, #3 and #4 respectively. These vectors are associated with TIMER 2 (#1), VSYNC (#2), TIMER 1 (#3) and PC4(PWRIN) (#4).

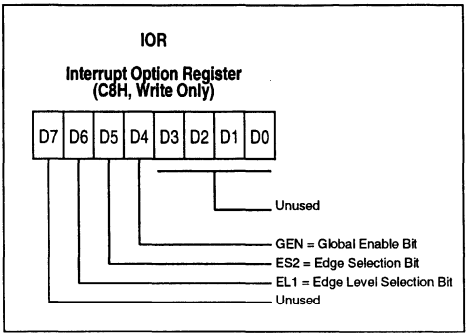
Interrupt Priority

The non-maskable interrupt request has the highest priority and can interrupt any other interrupt routines at any time, nevertheless the other interrupts cannot interrupt each other. If more than one interrupt request is pending, they are processed by the ST6369 Core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is hardware fixed.

Interrupt Option Register

The Interrupt Option Register (IOR register, location C8H) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register can be addressed in the Data Space as RAM location at the C8H address, nevertheless it is write-only register that can not be accessed with single-bit operations. The operating modes of the external interrupt inputs associated to interrupt vectors #1 and #2 are selected through bits 4 and 5 of the IOR register.

Figure 18. Interrupt Option Register



D7. Not used.

EL1. This is the Edge/Level selection bit of interrupt #1. When set to one, the interrupt is generated on low level of the related signal; when cleared to zero, the interrupt is generated on falling edge. The bit is cleared to zero after reset.

ES2. This is the edge selection bit on interrupt #2. This bit is used on the ST6369 devices with on-chip OSD generator for VSYNC detection.

GEN. This is the global enable bit. When set to one all interrupts are globally enabled; when this bit is cleared to zero all interrupts are disabled (excluding NMI).

D3 - D0. These bits are not used.

INTERRUPT (Continued)

Interrupt Procedure

The interrupt procedure is very similar to a call procedure; the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event the user does not know about the context and the time at which it occurred. As a result the user should save all the data space registers which will be used inside the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes which are automatically switched and so these do not need to be saved.

The following list summarizes the interrupt procedure (refer also to Figure 19. Interrupt Processing Flow Chart):

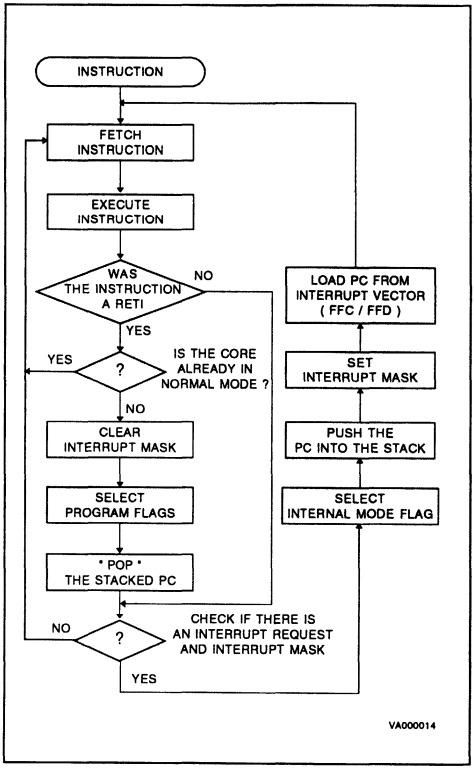
- Interrupt detection
- The flags C and Z of the main routine are exchanged with the flags C and Z of the interrupt routine (resp. the NMI flags)
- The value of the PC is stored in the first level of the stack - The normal interrupt lines are inhibited (NMI still active)
- The edge flip-flop is reset
- The related interrupt vector is loaded in the PC.
- User selected registers are saved inside the interrupt service routine (normally on a software stack)
- The source of the interrupt is found by polling (if more than one source is associated to the same vector)
- Interrupt servicing
- Return from interrupt (RETI)
- Automatically the ST63xx core switches back to the normal flags (resp the interrupt flags) and pops the previous PC value from the stack

The interrupt routine begins usually by the identification of the device that has generated the interrupt request. The user should save the registers which are used inside the interrupt routine (that holds relevant data) into a software stack.
After the RETI instruction execution, the Core carries out the previous actions and the main routine can continue.

ST6369 Interrupt Details

IR Interrupt (#0). The IRIN/PC6 Interrupt is connected to the first interrupt #0 (NMI, 0FFCH). If the $\overline{\text{IRINT}}$ interrupt is disabled at the Latch circuitry, then it will be high. The #0 interrupt input detects a high to low level. Note that once #0 has been latched, then the only way to remove the latched #0 signal is to service the interrupt. #0 can interrupt the other interrupts. A simple latch is provided from the PC6(IRIN) pin in order to generate the $\overline{\text{IRINT}}$ signal. This latch can be triggered by either the positive or negative edge of IRIN signal. $\overline{\text{IRINT}}$ is inverted with respect to the latch. The latch can be read by software and reset by software.

Figure 19. Interrupt Processing Flow-Chart



INTERRUPT (Continued)

TIMER 2 Interrupt (#1). The TIMER 2 Interrupt is connected to the interrupt #1 (0FF6H). The TIMER 2 interrupt generates a low level (which is latched in the timer). Only the low level selection for #1 can be used. Bit 6 of the interrupt option register C8H has to be set.

VSYNC Interrupt (#2). The VSYNC Interrupt is connected to the interrupt #2. When disabled the VSYNC INT signal is low. The VSYNC INT signal is inverted with respect to the signal applied to the VSYNC pin. Bit 5 of the interrupt option register C8H is used to select the negative edge (ES2=0) or the positive edge (ES2=1); the edge will depend on the application. Note that once an edge has been latched, then the only way to remove the latched signal is to service the interrupt. Care must be taken not to generate spurious interrupts. This interrupt may be used for synchronize to the VSYNC signal in order to change characters in the OSD only when the screen is on vertical blanking (if desired). This method may also be used to blink characters.

TIMER 1 Interrupt (#3). The TIMER 1 Interrupt is connected to the fourth interrupt #3 (0FF2H) which detects a low level (latched in the timer).

PWR Interrupt (#4). The PWR Interrupt is connected to the fifth interrupt #4 (0FF0H). If the PWRINT is disabled at the PWR circuitry, then it will be high. The #4 interrupt input detects a low level. A simple latch is provided from the PC4 (PWRIN)pin in order to generate the PWRINT signal. This latch can be triggered by either the positive or negative edge of the PWRIN signal. PWRINT is inverted with respect to the latch. The latch can be reset by software.

Notes Global disable does not reset edge sensitive interrupt flags. These edge sensitive interrupts become pending again when global disabling is released. Moreover, edge sensitive interrupts are stored in the related flags also when interrupts are globally disabled, unless each edge sensitive interrupt is also individually disabled before the interrupting event happens. Global disable is done by clearing the GEN bit of Interrupt option register, while any individual disable is done in the control register of the peripheral. The on-chip Timer peripherals have an interrupt request flag bit (TMZ), this bit is set to one when the device wants to generate an interrupt request and a mask bit (ETI) that must be set to one to allow the transfer of the flag bit to the Core.

RESET

The ST6369 devices can be reset in two ways: by the external reset input (RESET) tied low and by the hardware activated digital watchdog peripheral.

RESET Input

The external active low reset pin is used to reset the ST6369 devices and provide an orderly software startup procedure. The activation of the Reset pin may occur at any time in the RUN or WAIT mode. Even short pulses at the reset pin will be accepted since the reset signal is latched internally and is only cleared after 2048 clocks at the oscillator pin. The clocks from the oscillator pin to the reset circuitry are buffered by a schmitt trigger so that an oscillator in start-up conditions will not give spurious clocks. When the reset pin is held low, the external crystal oscillator is also disabled in order to reduce current consumption. The MCU is configured in the Reset mode as long as the signal of the RESET pin is low. The processing of the program is stopped and the standard Input/Output ports (port A, port B and port C) are in the input state. As soon as the level on the reset pin becomes high, the initialization sequence is executed. Refer to the MCU initialization sequence for additional information.

Watchdog Reset

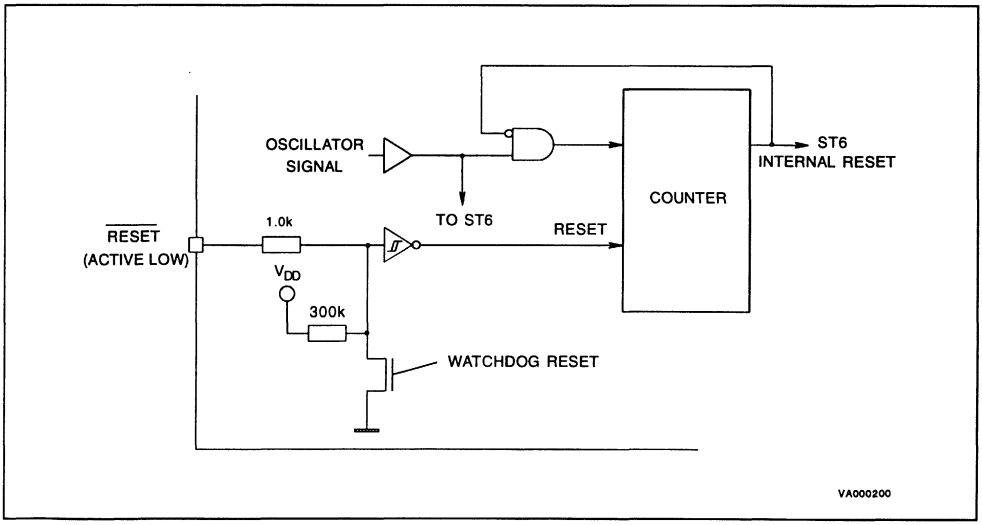
The ST6369 devices are provided with an on-chip hardware activated digital watchdog function in order to provide a graceful recovery from a software upset. If the watchdog register is not refreshed and the end-of-count is reached, then the reset state will be latched into the MCU and an internal circuit pulls down the reset pin. This also resets the watchdog which subsequently turns off the pull-down and activates the pull-up device at the reset pin. This causes the positive transition at the reset pin. The MCU will then exit the reset state after 2048 clocks on the oscillator pin.

Application Notes

An external resistor between V_{DD} and the reset pin is not required because an internal pull-up device is provided. The user may prefer to add an external pull-up resistor.

An internal Power-on device does not guarantee that the MCU will exit the reset state when V_{DD} is above 4.5V and therefore the RESET pin should be externally controlled.

Figure 20. Internal Reset Circuit



RESET (Continued)

Figure 21. Reset & Interrupt Processing Flow-Chart

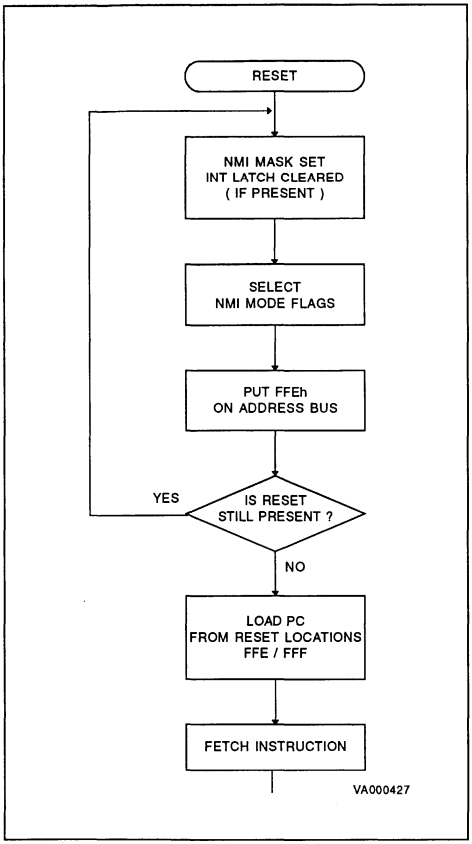
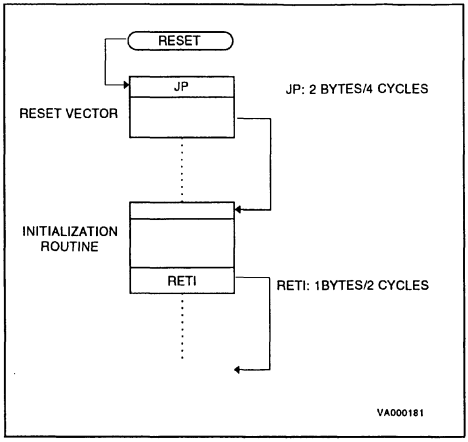


Figure 22. Restart Initialization Program Flow-Chart



MCU Initialization Sequence

When a reset occurs the stack is reset to program counter, the PC is loaded with the address of the reset vector (located in the program ROM at addresses FFEH & FFFH). A jump instruction to the beginning of the program has to be written into these locations.After a reset the interrupt mask is automatically activated so that the Core is in non-maskable interrupt mode to prevent false or ghost interrupts during the restart phase. Therefore the restart routine should be terminated by a RETI instruction to switch to normal mode and enable interrupts. If no pending interrupt is present at the end of the reset routine, the ST6369 will continue with the instruction after the RETI; otherwise the pending interrupt will be serviced.

RESET Low Power Mode

When the reset pin is low, the quartz oscillator is Disabled allowing reduced current consumption. When the reset pin is raised the quartz oscillator is enabled and oscillations will start to build up.The internal reset circuitry will count 2048 clocks on the oscillator pin before allowing the MCU to go out of the reset state;the clocks are after a schmitt trigger so that false or multiple counts are not possible.

WAIT & STOP MODES

The STOP and WAIT modes have been implemented in the ST6369 Core in order to reduce the consumption of the device when the latter has no instruction to execute. These two modes are described in the following paragraphs. On ST6369 as the hardware activated digital watchdog function is present the STOP instruction is de-activated and any attempt to execute it will cause the automatic execution of a WAIT instruction.

WAIT Mode

The configuration of the MCU in the WAIT mode occurs as soon as the WAIT instruction is executed. The microcontroller can also be considered as being in a "software frozen" state where the Core stops processing the instructions of the routine, the contents of the RAM locations and peripheral registers are saved as long as the power supply voltage is higher than the RAM retention voltage but where the peripherals are still working.

The WAIT mode is used when the user wants to reduce the consumption of the MCU when it is in idle, while not losing count of time or monitoring of external events. The oscillator is not stopped in order to provide clock signal to the peripherals. The timers counting may be enabled (writing the PSI bit in TSCR register) and the timer interrupt may be also enabled before entering the WAIT mode; this allows the WAIT mode to be left when timer interrupt occurs. If the exit from the WAIT mode is performed with a general RESET (either from the activation of the external pin or by watchdog reset) the MCU will enter a normal reset procedure as described in the RESET chapter. If an interrupt is generated during WAIT mode the MCU behaviour depends on the state of the ST6369 Core before the initialization of the WAIT sequence, but also of the kind of the interrupt request that is generated. This case will be described in the following paragraphs. In any case, the ST6369 Core does not generate any delay after the occurrence of the interrupt because the oscillator clock is still available.

STOP Mode

On ST6369 the hardware watchdog is present and the STOP instruction has been de-activated. Any attempt to execute a STOP will cause the automatic execution of a WAIT instruction.

Exit from WAIT Mode

The following paragraphs describe the output procedure of the ST6369 Core from WAIT mode when an interrupt occurs. It must be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) before the start of the WAIT sequence, but also of the type of the interrupt request that is generated. In all cases the GEN bit of IOR has to be set to 1 in order to restart from WAIT mode. Contrary to the operation of NMI in the run mode, the NMI is masked in WAIT mode if GEN=0.

Normal Mode. If the ST6369 Core was in the main routine when the WAIT instruction has been executed, the ST6369 Core outputs from the wait mode as soon as any interrupt occurs; the related interrupt routine is executed and at the end of the interrupt service routine the instruction that follows the WAIT instruction is executed if no other interrupts are pending.

Non-maskable Interrupt Mode. If the WAIT instruction has been executed during the execution of the non-maskable interrupt routine, the ST6369 Core outputs from the wait mode as soon as any interrupt occurs: the instruction that follows the WAIT instruction is executed and the ST6369 Core is still in the non-maskable interrupt mode even if another interrupt has been generated.

Normal Interrupt Mode. If the ST6369 Core was in the interrupt mode before the initialization of the WAIT sequence, it outputs from the wait mode as soon as any interrupt occurs. Nevertheless, two cases have to be considered:

- If the interrupt is a normal interrupt, the interrupt routine in which the WAIT was entered will be completed with the execution of the instruction that follows the WAIT and the ST6369 Core is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance to their priority.
- If the interrupt is a non-maskable interrupt, the non-maskable routine is processed at first. Then, the routine in which the WAIT was entered will be completed with the execution of the instruction that follows the WAIT and the ST6369 Core is still in the normal interrupt mode.

Notes :

If all the interrupt sources are disabled, the restart of the MCU can only be done by a Reset activation. The Wait instruction is not executed if an enabled interrupt request is pending. In the ST6369 the hardware activated digital watchdog function is present. As the watchdog is always activated the STOP instruction is de-activated and any attempt to execute the STOP instruction will cause an execution of a WAIT instruction.

ON-CHIP CLOCK OSCILLATOR

The internal oscillator circuit is designed to require a minimum of external components. A crystal quartz, a ceramic resonator, or an external signal (provided to the OSCIN pin) may be used to generate a system clock with various stability/cost tradeoffs. The typical clock frequency is 8MHz. Please note that different frequencies will affect the operation of those peripherals (D/As, SPI) whose reference frequencies are derived from the system clock.

The different clock generator options connection methods are shown in Figures 23 and 24. One machine cycle takes 13 oscillator pulses; 12 clock pulses are needed to increment the PC while and additional 13th pulse is needed to stabilize the internal latches during memory addressing. This means that with a clock frequency of 8MHz the machine cycle is 1.625µSec.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially RS), oscillator load capacitance (CL), IC parameters, ambient temperature, and supply voltage.It must be observed that the crystal or ceramic leads and circuit connections must be as short as possible. Typical values for CL1 and CL2 are in the range of 15pF to 22pF but these should be chosen based on the crystal manufacturers specification. Typical input capacitance for OSCIN and OSCOUT pins is 5pF.

The oscillator output frequency is internally divided by 13 to produce the machine cycle and by 12 to produce the Timer and the Watchdog clock. A byte cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five byte cycles to be executed (See Table 7).

Table 7. Intructions Timing with 8MHz Clock

Instruction Type	Cycles	Execution Time
Branch if set/reset	5 Cycles	8.125µs
Branch & Subroutine Branch	4 Cycles	6.50µs
Bit Manipulation	4 Cycles	6.50µs
Load Instruction	4 Cycles	6.50µs
Arithmetic & Logic	4 Cycles	6.50µs
Conditional Branch	2 Cycles	3.25µs
Program Control	2 Cycles	3.25µs

Figure 23. Clock Generator Option (1)

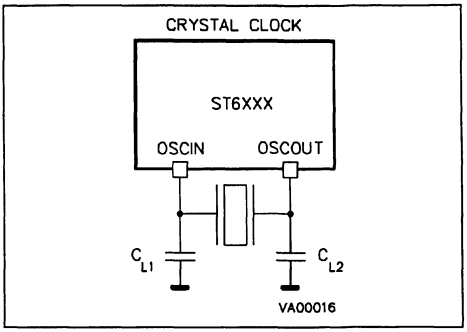


Figure 24. Clock Generator Option (2)

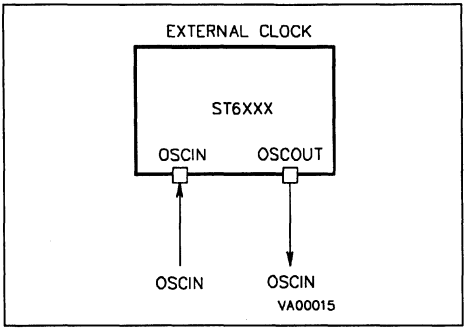
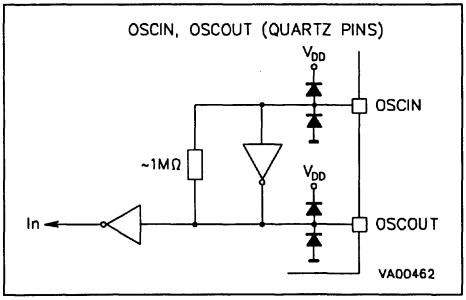


Figure 25. OSCIN, OSCOUT Diagram



INPUT/OUTPUT PORTS

The ST6369 microcontrollers use three standard I/O ports (A,B,C) with up to eight pins on each port; refer to the device pin configurations to see which pins are available.

Each line can be individually programmed either in the input mode or the output mode as follows by software.

- Output
- Input with on-chip pull-up resistor (selected by software)
- Input without on-chip pull-up resistor (selected by software)

Note: pins with 12V open-drain capability do not have pull-up resistors.

In output mode the following hardware configurations are available:

- Open-drain output 12V (PA4-PA7, PC4-PC7)
- Open-drain output 5V (PC0-PC3)
- Push-pull output (PA0-PA3, PB0-PB6)

The lines are organized in three ports (port A,B,C). The ports occupy 6 registers in the data space. Each bit of these registers is associated with a particular line (for instance, the bits 0 of the Port A Data and Direction registers are associated with the PA0 line of Port A).

There are three Data registers (DRA, DRB, DRC), that are used to read the voltage level values of the lines programmed in the input mode, or to write the logic value of the signal to be output on the lines configured in the output mode. The port Data Registers can be read to get the effective logic levels of the pins, but they can be also written by the user software, in conjunction with the related Data Direction Register, to select the different input mode options. Single-bit operations on I/O registers (bit set/reset instructions) are possible but care is necessary because reading in input mode is made from I/O pins and therefore might be influenced by the external load, while writing will directly affect the Port data register causing an undesired changes of the input configuration. The three Data Direction registers (DDRA, DDRB, DDRC) allow the selection of the direction of each pin (input or output).

All the I/O registers can be read or written as any other RAM location of the data space, so no extra RAM cell is needed for port data storing and manipulation. During the initialization of the MCU, all the I/O registers are cleared and the input mode with pull-up is selected on all the pins thus avoiding pin conflicts (with the exception of PC2 that is set in output mode and is set high ie. high impedance).

Details of I/O Ports

When programmed as an input a pull-up resistor (if available) can be switched active under program control. When programmed as an output the I/O port will operate either in the push-pull mode or the open-drain mode according to the hardware fixed configuration as specified below.

Port A. PA0-PA3 are available as push-pull when outputs. PA4-PA7 are available as open-drain (no push-pull programmability) capable of withstanding 12V (no resistive pull-up in input mode). PA6-PA7 has been specially designed for higher driving capability and are able to sink 25mA with a maximum V_{OL} of 1V.

Port B. All lines are configured as push-pull when outputs.

Port C. PC0-PC3 are available as open-drain capable of withstanding a maximum V_{DD}+0.3V. PC4-PC7 are available as open-drain capable of withstanding 12V (no resistive pull-up in input mode). Some lines are also used as I/O buffers for signals coming from the on-chip SPI.

In this case the final signal on the output pin is equivalent to a wired AND with the programmed data output.

If the user needs to use the serial peripheral, the I/O line should be set in output mode while the open-drain configuration is hardware fixed; the corresponding data bit must set to one. If the latched interrupt functions are used (HSYNC, PWRIN) then the corresponding pins should be set to input mode.

On ST6369 the I/O pins with double or special functions are:

- PC0/SCL (connected to the SPI clock signal)
- PC1/SDA (connected to the SPI data signal)
- PC3/SEN (connected to the SPI enable signal)
- PC4/PWRIN (connected to the PWRIN interrupt latch)
- PC6/HSYNC (connected to the HSYNC interrupt latch)

All the Port A,B and C I/O lines have Schmitt-trigger input configuration with a typical hysteresis of 1V.

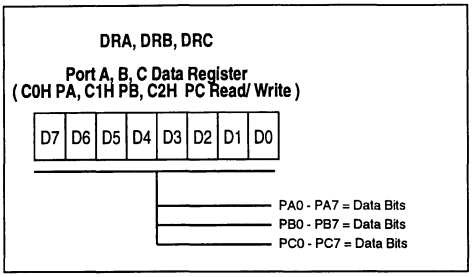
INPUT/OUTPUT PORTS (Continued)

Table 8. I/O Port Options Selection

DDR	DR	Mode	Option
0	0	Input	With on-chip pull-up resistor
0	1	Input	Without on-chip pull-up resistor
1	X	Output	Open-drain or Push-Pull

Note: X: Means don't care.

Figure 26. Port A, B, C Data Register



PA7-PA0. These are the I/O port A data bits. Reset at power-on.

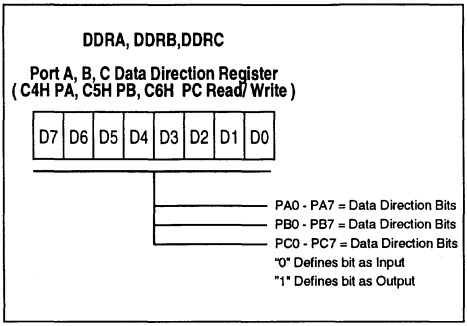
PB7-PB0. These are the I/O port B data bits. Reset at power-on.

PC7-PC0. Set to 04H at power-on. Bit 2 (PC2 pin) is set to one (open drain therefore high impedence).

I/O Pin Programming

Each pin can be individually programmed as input or output with different input and output configurations. This is achieved by writing to the relevant bit in the data (DR) and data direction register (DDR). Table 8 shows all the port configurations that can be selected by the user software.

Figure 27. Port A, B, C Data Register



PA7-PA0. These are the I/O port A data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Reset at power-on.

PB7-PB0. These are the I/O port B data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Reset at power-on.

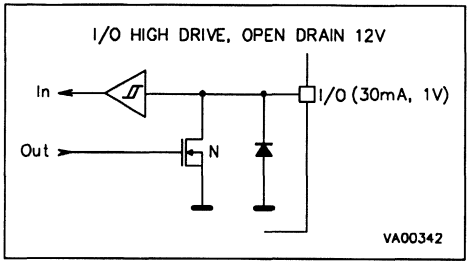
PC7-PC0. These are the I/O port C data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Set to 04H at power-on. Bit 2 (PC2 pin) is set to one (output mode selected).

INPUT/OUTPUT PORTS (Continued)

Input/Output Configurations

The following schematics show the I/O lines hardware configuration for the different options. Figure 28 shows the I/O configuration for an I/O pin with open-drain 12V capability (standard drive and high drive). Figure 29 shows the I/O configuration for an I/O pin with push-pull and with open drain 5V capability.

Figure 28. I/O Configuration Diagram (Open Drain 12V)

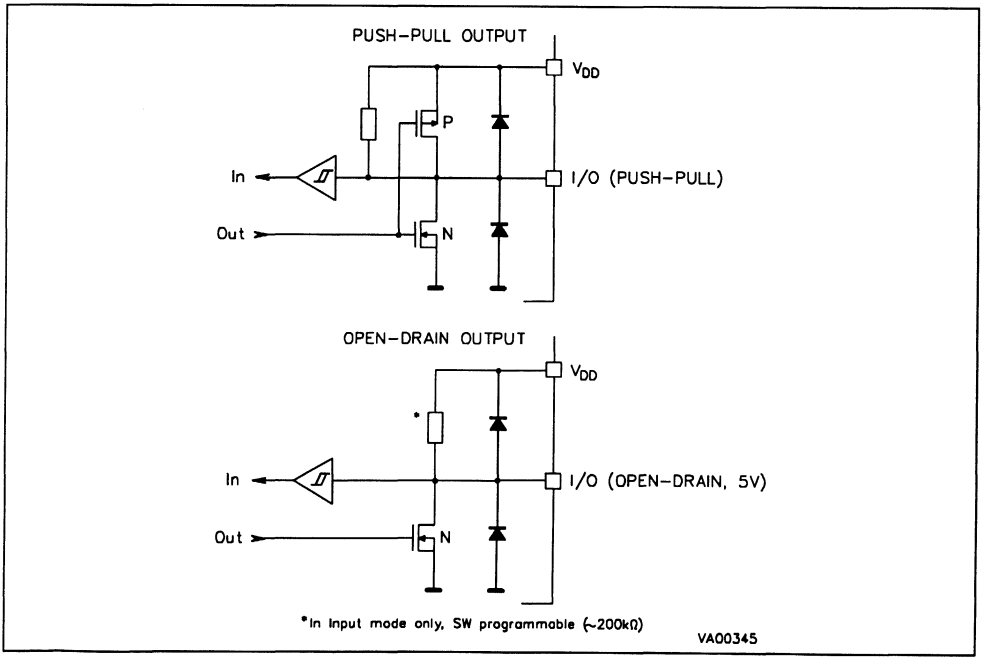


Notes :

The WAIT instruction allows the ST6369 to be used in situations where low power consumption is needed. This can only be achieved however if the I/O pins either are programmed as inputs with well defined logic levels or have no power consuming resistive loads in output mode. The unavailable I/O lines PB0, PB3 and PB7 should be programmed in output mode.

Single-bit operations on I/O registers are possible but **care is necessary** because reading in input mode is made from I/O pins while writing will directly affect the Port data register causing an undesired changes of the input configuration.

Figure 29. I/O Configuration Diagram (Open Drain 5V, Push-pull)



TIMERS

The ST6369 devices offer two on-chip Timer peripherals consisting of an 8-bit counter with a 7-bit programmable prescaler, thus giving a maximum count of 2^{15}, and a control logic that allows configuring the peripheral operating mode. Figure 30 shows the timer block diagram. The content of the 8-bit counters can be read/written in the Timer/Counter registers TCR that can be addressed in the data space as RAM location at addresses D3H (Timer 1) and DBH (Timer 2). The state of the 7-bit prescaler can be read in the PSC register at addresses D2H (Timer 1) and DAH (Timer 2). The control logic is managed by TSCR registers at D4H (Timer 1) and DCH (Timer 2) addresses as described in the following paragraphs.

The following description applies to both Timer 1 and Timer 2. The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (timer zero) bit in the TSCR is set to one. If the ETI (enable timer interrupt) bit in the TSCR is also set to one an interrupt request, associated to interrupt vector #3 (for Timer 1) and #1 for Timer 2, is generated. The interrupt of the timer can be used to exit the MCU from the WAIT mode.

The prescaler decrements on rising edge. The prescaler input is the oscillator frequency divided by 12.

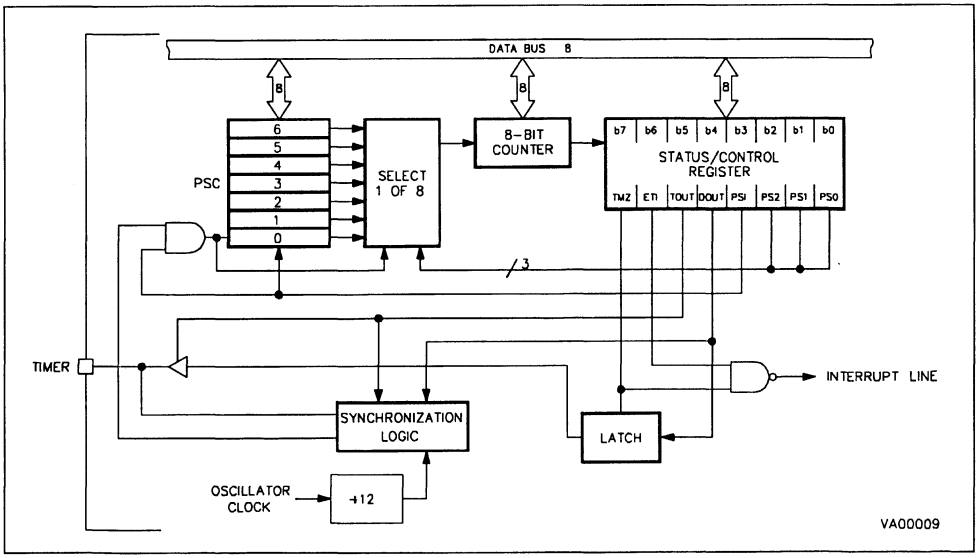
Depending on the division factor programmed by PS2/PS1/PS0 (see table 9) bits in the TSCR, the clock input of the timer/counter register is multiplexed to different sources.

On division factor 1, the clock input of the prescaler is also that of timer/counter; on factor 2, bit 0 of prescaler register is connected to the clock input of TCR.

This bit changes its state with the half frequency of prescaler clock input. On factor 4, bit 1 of PSC is connected to clock input of TCR, and so on. On division factor 128, the MSB bit 6 of PSC is connected to clock input of TCR. The prescaler initialize bit (PSI) in the TSCR register must be set to one to allow the prescaler (and hence the counter) to start. If it is cleared to zero then all of the prescaler bits are set to one and the counter is inhibited from counting.

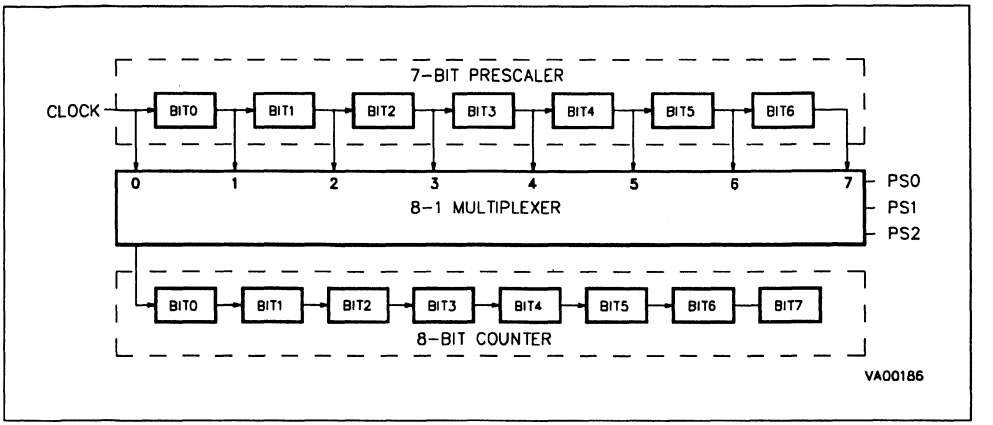
The prescaler can be given any value between 0 and 7FH by writing to the related register address, if bit PSI in the TSCR register is set to one. The tap of the prescaler is selected using the PS2/PS1/PS0 bits in the control register. Figure 31 shows the timer working principle.

Figure 30. Timer Peripheral Block Diagram



TIMERS (Continued)

Figure 31. Timer Working Principle



Timer Operating Modes

As on ST6369 devices the external TIMER pin is not available the only allowed operating mode is the output mode that have to be selected by setting to 1 bit 4 and by clearing to 0 bit 5 in the TSCR1 register. This procedure will enable both Timer 1 and Timer 2.

Output Mode (TSCR1 D4 = 1, TSCR1 D5 = 0). On this mode the timer prescaler is clocked by the prescaler clock input (OSC/12). The user can select the desired prescaler division ratio through the PS2/PS1/PS0 bits. When TCR count reaches 0, it sets the TMZ bit in the TSCR.

The TMZ bit can be tested under program control to perform timer functions whenever it goes high. Bit D4 and D5 on TSCR2 (Timer 2) register are not implemented.

Timer Interrupt

When the counter register decrements to zero and the software controlled ETI (enable timer interrupt) bit is set to one then an interrupt request associated to interrupt vector #3 (for Timer 1) and to interrupt vector #1 (for Timer 2) is generated. When the counter decrements to zero also the TMZ bit in the TSCR register is set to one.

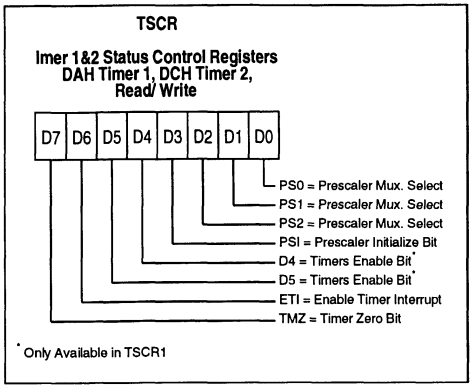
Notes :

TMZ is set when the counter reaches 00H ; however, it may be set by writing 00H in the TCR register or setting the bit 7 of the TSCR register. TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded to FFH while the 7-bit prescaler is loaded to 7FH , and the TSCR register is cleared which means that timer is stopped (PSI=0) and timer interrupt disabled.

A write to the TCR register will predominate over the 8-bit counter decrement to 00H function, i.e. if a write and a TCR register decrement to 00H occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00H again. The values of the TCR and the PSC registers can be read accurately at any time.

TIMERS (Continued)

Figure 32. Timer Status Control Registers



TMZ. Low-to-high transition indicates that the timer count register has decrement to zero. This bit must be cleared by user software before to start with a new count.

ETI. This bit, when set, enables the timer interrupt (vector #3 for Timer 1, vector #1 for Timer 2) request. If ETI=0 the timer interrupt is disabled. If ETI= 1 and TMZ= 1 an interrupt request is generated.

D5. This is the timers enable bit D5. It must be cleared to 0 together with a set to 1 of bit D4 to enable both Timer 1 and Timer 2 functions. It is not implemented on TSCR2 register.

D4. This is the timers enable bit D4. This bit must be set to 1 together with a clear to 0 of bit D5 to enable both Timer 1 and Timer 2 functions. It is not implemented on TSCR2 register.

D5	D4	Timers
0	0	Disabled
0	1	Enabled
1	X	Reserved

PS1. Used to initialize the prescaler and inhibit its counting while PSI = 0 the prescaler is set to 7FH and the counter is inhibited. When PSI = 1 the prescaler is enabled to count downwards. As long as PSI= 0 both counter and prescaler are not running.

PS2-PS0. These bits select the division ratio of the prescaler register. (see table 9)

The TSCR1 and TSCR2 registers are cleared on reset. The correct D4-D5 combination must be written in TSCR1 by user's software to enable the operation of Timer 1 and Timer 2.

Table 9. Prescaler Division Factors

PS2	PS1	PS0	Divided By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Figure 33. Timer Counter Registers

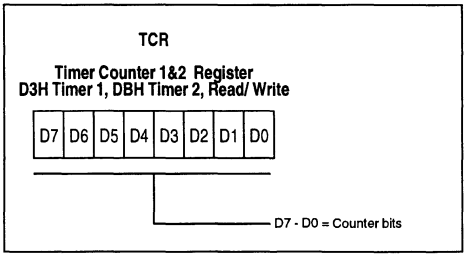
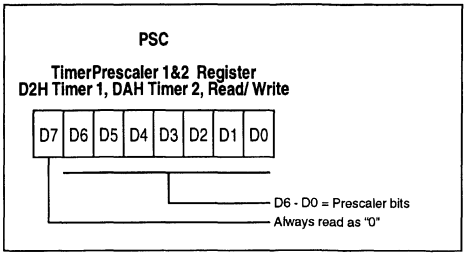


Figure 34. Timer Counter Registers



HARDWARE ACTIVATED DIGITAL WATCHDOG FUNCTION

The hardware activated digital watchdog function consists of a down counter that is automatically initialized after reset so that this function does not need to be activated by the user program. As the watchdog function is always activated this down counter can not be used as a timer. The watchdog is using one data space register (HWDR location D8H). The watchdog register is set to FEH on reset and immediately starts to count down, requiring no software start. Similarly the hardware activated watchdog can not be stopped or delayed by software.

The watchdog time can be programmed using the 6 MSbits in the watchdog register, this gives the possibility to generate a reset in a time between 3072 to 196608 oscillator cycles in 64 possible steps. (With a clock frequency of 8MHz this means from 384µs to 24.576ms). The reset is prevented if the register is reloaded with the desired value before bits 2-7 decrement from all zeros to all ones.

The presence of the hardware watchdog deactivates the STOP instruction and a WAIT instruction is automatically executed instead of a STOP. Bit 1 of the watchdog register (set to one at reset) can be used to generate a software reset if cleared to zero). Figure 35 shows the watchdog block diagram while Figure 36 shows its working principle.

Figure 36. Hardware Activated Watchdog Working Principle

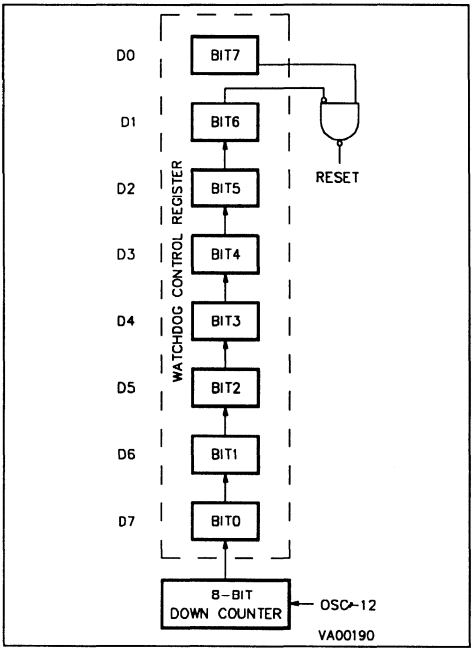
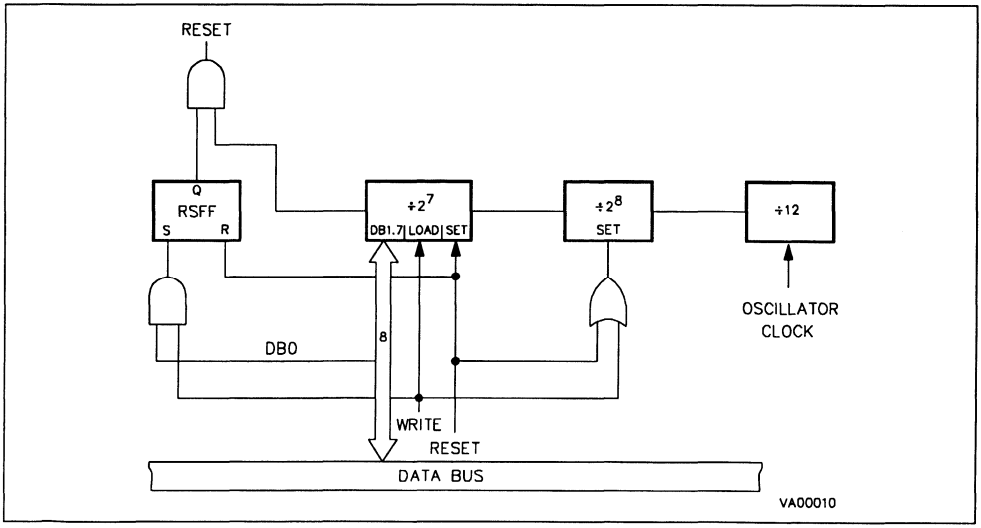
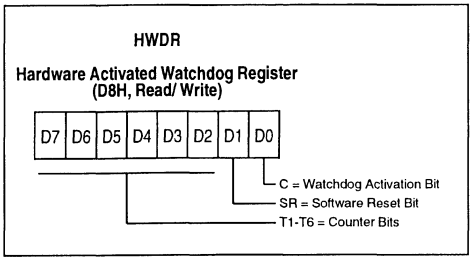


Figure 35. Hardware Activated Watchdog Block Diagram



HARDWARE ACTIVATED DIGITAL WATCHDOG FUNCTION (Continued)

Figure 37. Watchdog Register



T1-T6. These are the watchdog counter bits. It should be noted that D7 (T1) is the LSB of the counter and D2 (T6) is the MSB of the counter, these bits are in the opposite order to normal.

SR. This bit is set to one during the reset phase and will generate a software reset if cleared to zero.

C. This is the watchdog activation bit that is hardware set. The watchdog function is always activated independently of changes of value of this bit.

The register reset value is FEH (Bit 1-7 set to one, Bit 0 cleared).

SERIAL PERIPHERAL INTERFACE

The ST6369 Serial Peripheral Interface (SPI) has been designed to be cost effective and flexible in interfacing the various peripherals in TV applications.

It maintains the software flexibility but adds hardware configurations suitable to drive devices which require a fast exchange of data. The three pins dedicated for serial data transfer (single master only) can operate in the following ways:

- as standard I/O lines (software configuration)
- as S-BUS or as I^2CBUS (two pins)
- as standard (shift register) SPI

When using the hardware SPI, a fixed clock rate of 62.5kHz is provided.

It has to be noted that the first bit that is output on the data line by the 8-bit shift register is the MSB.

SPI Data/Control Registers

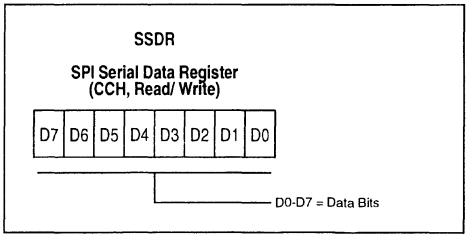
For I/O details on SCL (Serial Clock), SDA (Serial Data) and SEN (Serial Enable) please refer to I/O Ports description with reference to the following registers:

Port C data register, Address C2H (Read/Write).

- BIT D0 "SCL"
- BIT D1 "SDA"
- BIT D3 "SEN"

Port C data direction register, Address C6H (Read/Write).

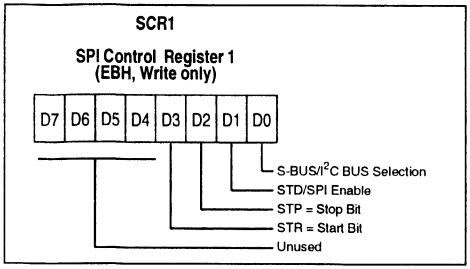
Figure 38. SPI Serial Data Register



D7-D0. These are the SPI data bits. They can be neither read nor written when SPI is operating (BUSY bit set). They are undefined after reset.

SERIAL PERIPHERAL INTERFACE (Continued)

Figure 39. SPI Control Register 1



D7-D4. These bits are not used.

STR. This is Start bit for I²CBUS/S-BUS. This bit is meaningless when STD/SPI enable bit is cleared to zero. If this bit is set to one STD/SPI bit is also set to "1" and SPI Start generation, before beginning of transmission, is enabled. Set to zero after reset.

STP. This is Stop bit for I²CBUS/S-BUS. This bit is meaningless when STD/SPI enable bit is cleared to zero. If this bit is set to one STD/SPI bit is also set to "1" and SPI Stop condition generation is enabled. STP bit must be reset when standard protocol is used (this is also the default reset conditions). Set to zero after reset.

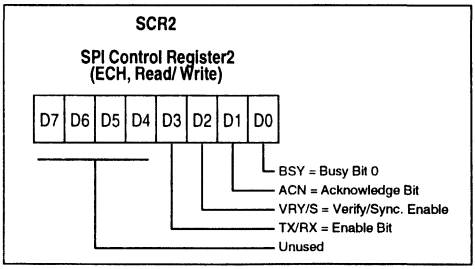
STD, SPI Enable. This bit, in conjunction with S-BUS/I²CBUS bit, allows the SPI disable and will select between I²CBUS/S-BUS and Standard shift register protocols. If this bit is set to one, it selects both I²CBUS and S-BUS protocols; final selection between them is made by S-BUS/I²CBUS bit. If this bit is cleared to zero when S-BUS/I²CBUS is set to "1" the Standard shift register protocol is selected. If this bit is cleared to "0" when S-BUS/I²CBUS is cleared to 0 the SPI is disabled. Set to zero after reset.

S-BUS/I²CBUS Selection. This bit, in conjunction with STD/SPI bit, allows the SPI disable and will select between I²CBUS and S-BUS protocols. If this bit is cleared to "0" when STD bit is also "0", the SPI interface is disabled. If this bit is cleared to zero when STD bit is set to "1", the I²CBUS protocol will be selected. If this bit is set to "1" when STD bit is set to "1", the S-BUS protocol will be selected. Cleared to zero after reset.

Table 10. SPI Modes Selection

D1 STD/SP	D0 S-BUS/I²C BUS	SPI Function
0	0	Disabled
0	1	STD Shift Reg.
1	0	I²C BUS
1	1	S-BUS

Figure 40. SPI Control Register2



D7-D4. These bits are not used.

TX/RX. *Write Only.* When this bit is set, current byte operation is a transmission. When it is reset, current operation is a reception. Set to zero after reset.

VRY/S. *Read Only/Write Only.* This bit has two different functions in relation to read or write operation. *Reading Operation:* when STD and/or TRX bits is cleared to 0, this bit is meaningless. When bits STD and TX are set to 1, this bit is set each time BSY bit is set. This bit is reset during byte operation if real data on SDA line are different from the output from the shift register. Set to zero after reset. *Writing Operation* : it enables (if set to one) or disables (if cleared to zero) the interrupt coming from VSYNC pin. Undefined after reset. Refer to OSD description for additional information.

ACN. *Read Only.* If STD bit (D1 of SCR1 register) is cleared to zero this bit is meaningless. When STD is set to one, this bit is set to one if no Acknowledge has been received. In this case it is automatically reset when BSY is set again. Set to zero after reset.

BSY. *Read/Set Only.* This is the busy bit. When a one is loaded into this bit the SPI interface start the transmission of the data byte loaded into SSDR data register or receiving and building the receive data into the SSDR data register. This is done in accordance with the protocol, direction and start/stop condition(s). This bit is automatically cleared at the end of the current byte operation. Cleared to zero after reset.

Note :
The SPI shift register is also the data transmission register and the data received register; this feature is made possible by using the serial structure of the ST6369 and thus reducing size and complexity.

SERIAL PERIPHERAL INTERFACE (Continued)

During transmission or reception of data, all access to serial data register is therefore disabled. The reception or transmission of data is started by setting the BUSY bit to "1"; this will be automatically reset at the end of the operation. After reset, the busy bit is cleared to "0", and the hardware SPI disabled by clearing bit 0 and bit 1 of SPI control register 1 to "0". The outputs from the hardware SPI are "ANDed" to the standard I/O software controlled outputs. If the hardware SPI is in operation the Port C pins related to the SPI should be configured as outputs using the Data Direction Register and should be set high. When the SPI is configured as the S-BUS, the three pins PC0, PC1 and PC3 become the pins SCL, SDA and SEN respectively. When configured as the I^2CBUS the pins PC0 and PC1 are configured as the pins SCL and SDA; PC3 is not driven and can be used as a general purpose I/O pin. In the case of the STD SPI the pins PC0 and PC1 become the signals CLOCK and DATA, PC3 is not driven and can be used as general purpose I/O pin. The VERIFY bit is available when the SPI is configured as either S-BUS or I^2CBUS. At the start of a byte transmission, the verify bit is set to one. If at any time during the transmission of the following eight bits, the data on the SDA line does not match the data forced by the SPI (while SCL is high), then the VERIFY bit is reset. The verify is available only during transmission for the S-BUS and I^2CBUS; for other protocol it is not defined. The SDA and SCL signal entering the SPI are buffered in order to remove any minor glitches. When STD bit is set to one (S-BUS or I^2CBUS selected), and TRX bit is reset (receiving data), and STOP bit is set (last byte of current communication), the SPI interface does not generate the Acknowledge, according to S-BUS/I^2CBUS specifications. PC0-SCL, PC1-SDA and PC3-SEN lines are standard drive I/O port pins with open-drain output configuration (maximum voltage that can be applied to these pins is V_{DD}+ 0.3V).

S-BUS/I^2CBUS Protocol Information

The S-BUS is a three-wire bidirectional data-bus with functional features similar to the I^2CBUS. In fact the S-BUS includes decoding of Start/Stop conditions and the arbitration procedure in case of multimaster system configuration (the ST6369 SPI allows a single-master only operation). The SDA line, in the I^2CBUS represents the AND combination of SDA and SEN lines in the S-BUS. If the SDA and the SEN lines are short-circuit connected, they appear as the SDA line of the I^2CBUS. The Start/Stop conditions are detected (by the external peripherals suited to work with S-BUS/I^2CBUS) in the following way:

- On S-BUS by a transition of the SEN line (1 to 0 Start, 0 to 1 Stop) while the SCL line is at high level.
- On I^2CBUS by a transition of the SDA line (10 Start, 01Stop) while the SCL line is at high level.

Start and Stop condition are always generated by the master (ST6369 SPI can only work as single master). The bus is busy after the start condition and can be considered again free only when a certain time delay is left after the stop condition. In the S-BUS configuration the SDA line is only allowed to change during the time SCL line is low. After the start information the SEN line returns to high level and remains unchanged for all the data transmission time. When the transmission is completed the SDA line is set to high level and, at the same time, the SEN line returns to the low level in order to supply the stop information with a low to high transition, while the SCL line is at high level. On the S-BUS, as on the I^2CBUS, each eight bit information (byte) is followed by one acknowledged bit which is a high level put on the SDA line by the transmitter. A peripheral that acknowledges has to pull down the SDA line during the acknowledge clock pulse. An addressed receiver has to generate an acknowledge after the reception of each byte; otherwise the SDA line remains at the high level during the ninth clock pulse time. In this case the master transmitter can generate the Stop condition, via the SEN (or SDA in I^2CBUS) line, in order to abort the transfer.

SERIAL PERIPHERAL INTERFACE (Continued)

Start/Stop Acknowledge. The timing specs of the S-BUS protocol require that data on the SDA (only on this line for I^2CBUS) and SEN lines be stable during the "high" time of SCL. Two exceptions to this rule are foreseen and they are used to signal the start and stop condition of data transfer.

- On S-BUS by a transition of the SEN line (10 Start, 01 Stop) while the SCL line is at high level.
- On I^2CBUS by a transition of the SDA line (10 Start, 01 Stop) while the SCL line is at high level.

Data are transmitted in 8-bit groups; after each group, a ninth bit is interposed, with the purpose of acknowledging the transmitting sequence (the transmit device place a "1" on the bus, the acknowledging receiver a "0").

Interface Protocol. This paragraph deals with the description of data protocol structure. The interface protocol includes:

- A start condition

- A "slave chip address" byte, transmitted by the master, containing two different information:

a. the code identifying the device the master wants to address (this information is present in the first seven bits)

b. the direction of transmission on the bus (this information is given in the 8th bit of the byte); "0" means "Write", that is from the master to the slave, while "1" means "Read". The addressed slave must always acknowledge.

The sequence from, now on, is different according to the value of $R/\overline{W}$ bit.

1. $R/\overline{W}$ = "0" ($\overline{Write}$)

In all the following bytes the master acts as transmitter; the sequence follows with:

a. an optional data byte to address (if needed) the slave location to be written (it can be a word address in a memory or a register address, etc.).

b. a "data" byte which will be written at the address given in the previous byte.

c. further data bytes.

d. a STOP condition

A data transfer is always terminated by a stop condition generated from the master. The ST6369 peripheral must finish with a stop condition before another start is given. Figure 44 shows an example of write operation.

2. $R/\overline{W}$ = "1" (Read)

In this case the slave acts as transmitter and, therefore, the transmission direction is changed. In read mode two different conditions can be considered:

a. The master reads slave immediately after first byte. In this case after the slave address sent from the master with read condition enabled the master transmitter becomes master receiver and the slave receiver becomes slave transmitter.

b. The master reads a specified register or location of the slave. In this case the first sent byte will contain the slave address with write condition enabled, then the second byte will specify the address of the register to be read. At this moment a new start is given together with the slave address in read mode and the procedure will proceed as described in previous point "a".

SERIAL PERIPHERAL INTERFACE (Continued)

Figure 41.Master Transmit to Slave Receiver (Write Mode)

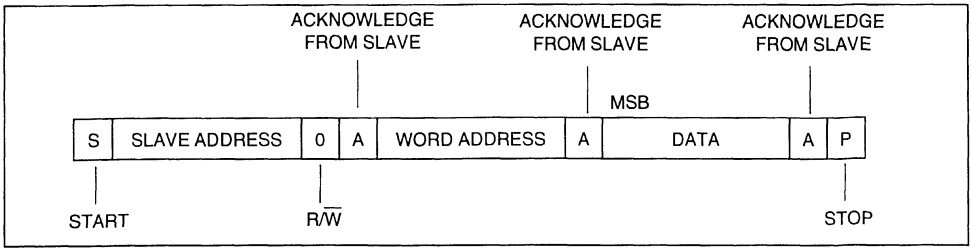


Figure 42.Master Reads Slave Immediately After First Byte (read Mode)

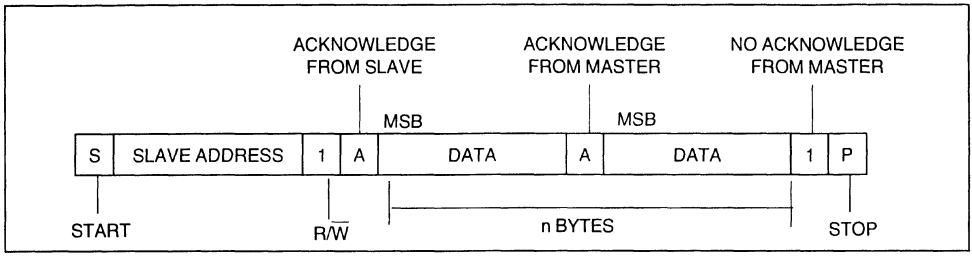
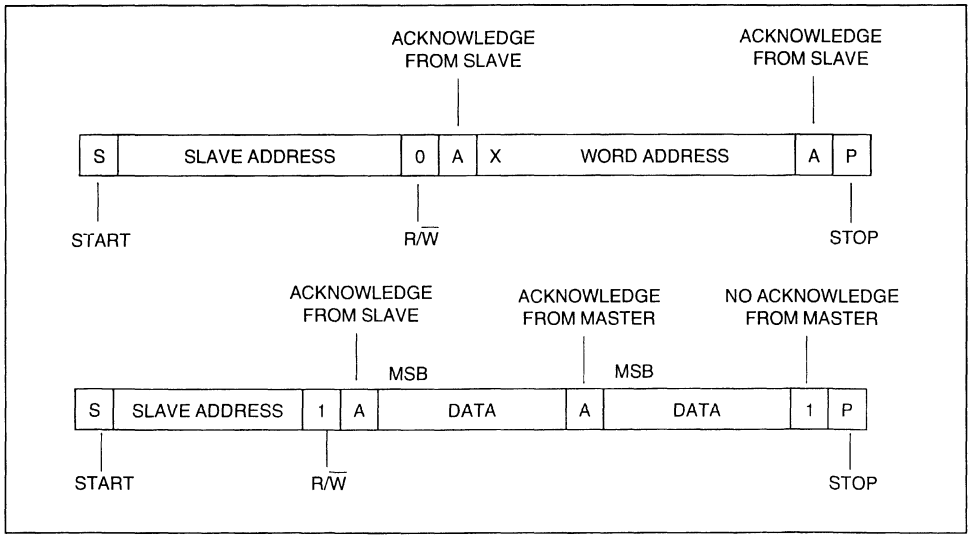


Figure 43.Master Reads After Setting Slave Register Address (Write Address, Read Data)

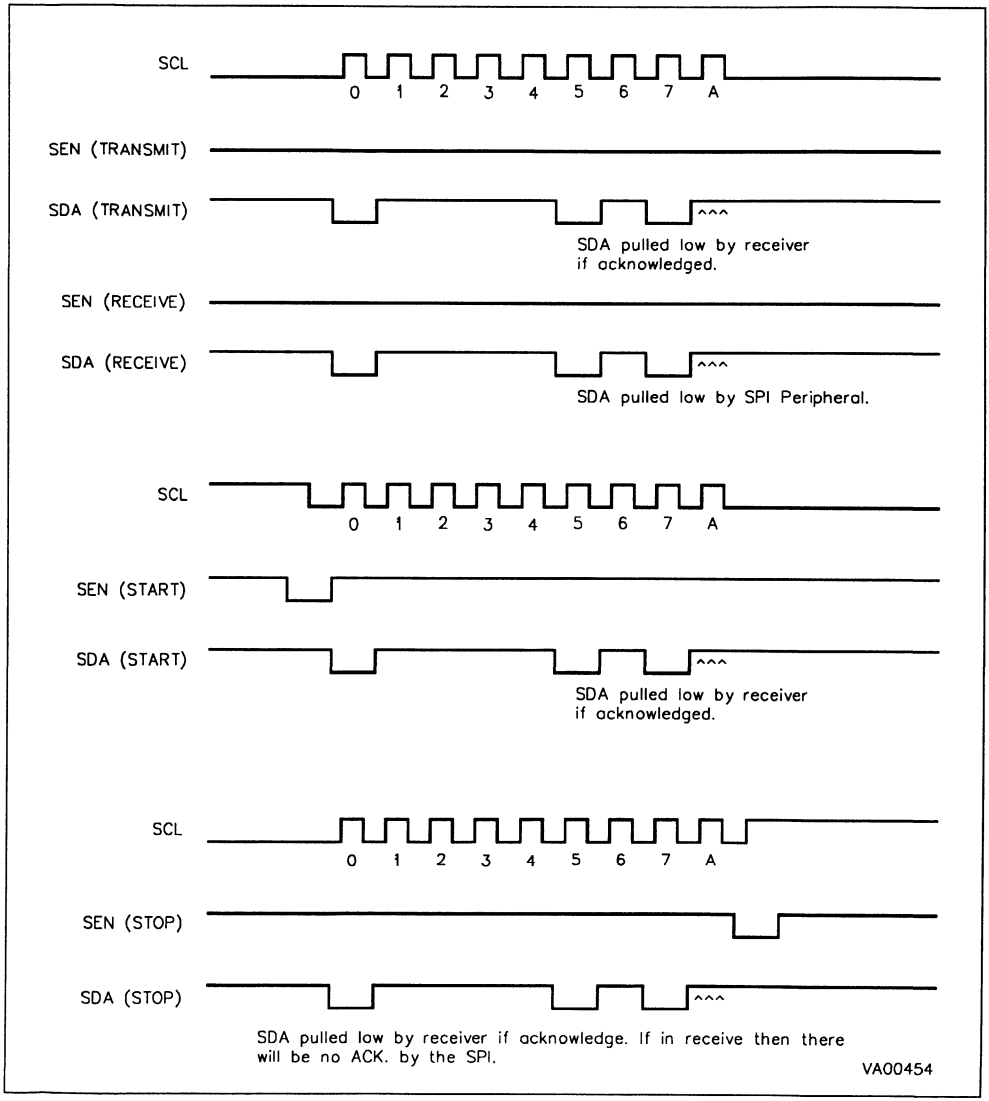


SERIAL PERIPHERAL INTERFACE (Continued)

S-BUS/I²CBUS Timing Diagrams

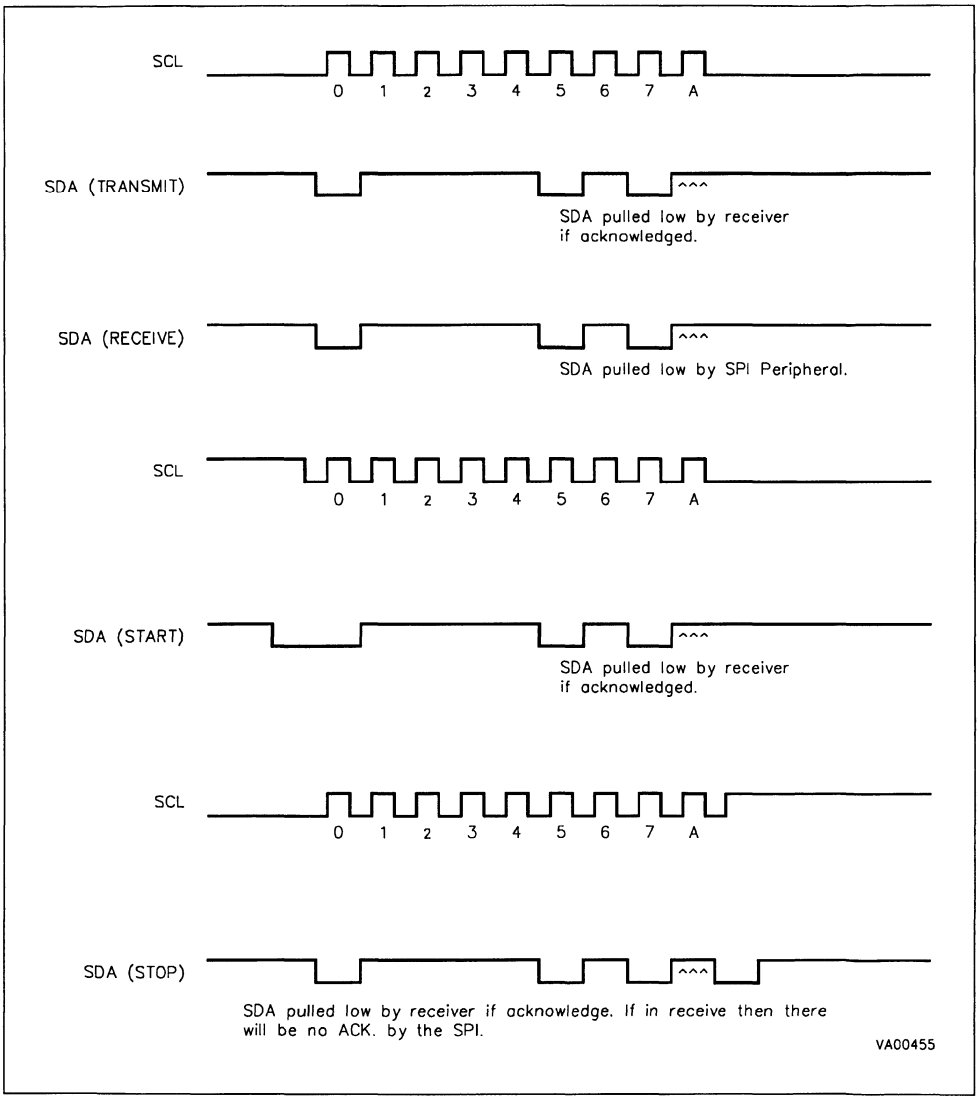
The clock of the S-BUS/I²CBUS of the ST6369 SPI (single master only) has a fixed bus clock frequency of 62.5KHz. All the devices connected to the bus must be able to follow transfers with frequencies up to 62.5KHz, either by being able to transmit or receive at that speed or by applying the clock synchronization procedure which will force the master into a wait state and stretch low periods.

Figure 44. S-BUS Timing Diagram



SERIAL PERIPHERAL INTERFACE (Continued)

Figure 45. I^2C BUS Timing Diagram



Note: The third pin, SEN, should be high; it is not used in the I^2CBUS. Logically SDA is the AND of the S-BUS SDA and SEN.

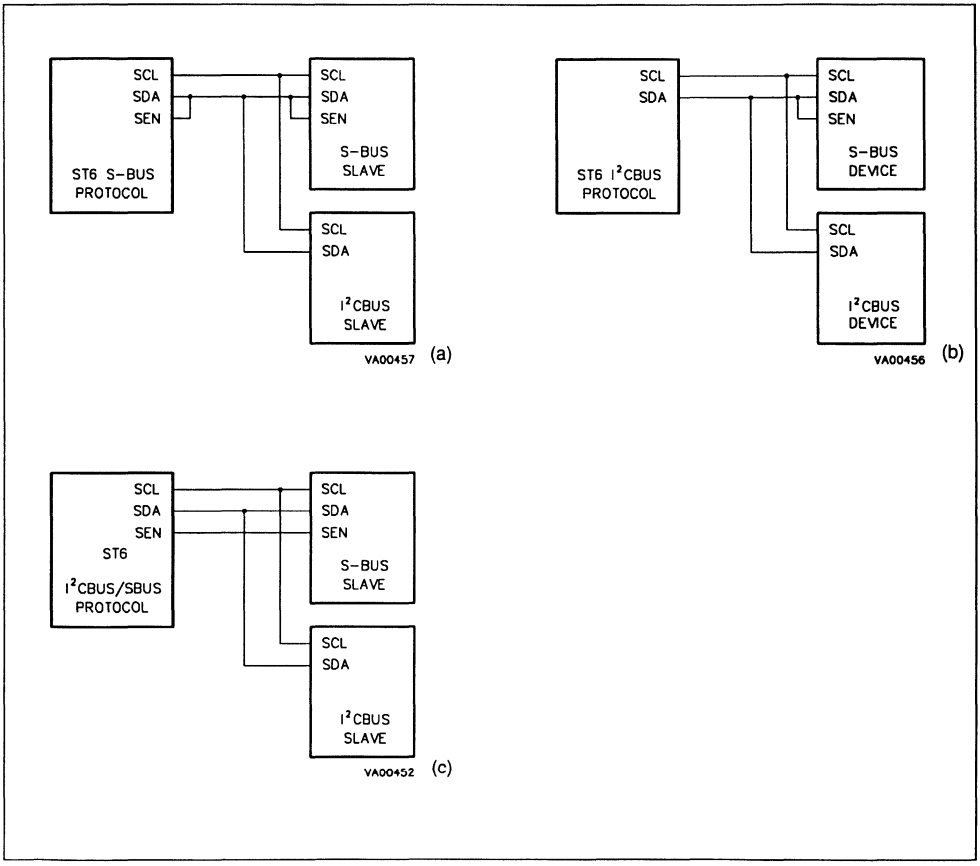
SERIAL PERIPHERAL INTERFACE (Continued)

Compatibility S-BUS/I²CBUS

Using the S-BUS protocol it is possible to implement mixed system including S-BUS/I²CBUS bus peripherals. In order to have the compatibility with the I²CBUS peripherals, the devices including the S-BUS interface must have their SDA and SEN pins connected together as shown in the following Figure 46 (a and b).

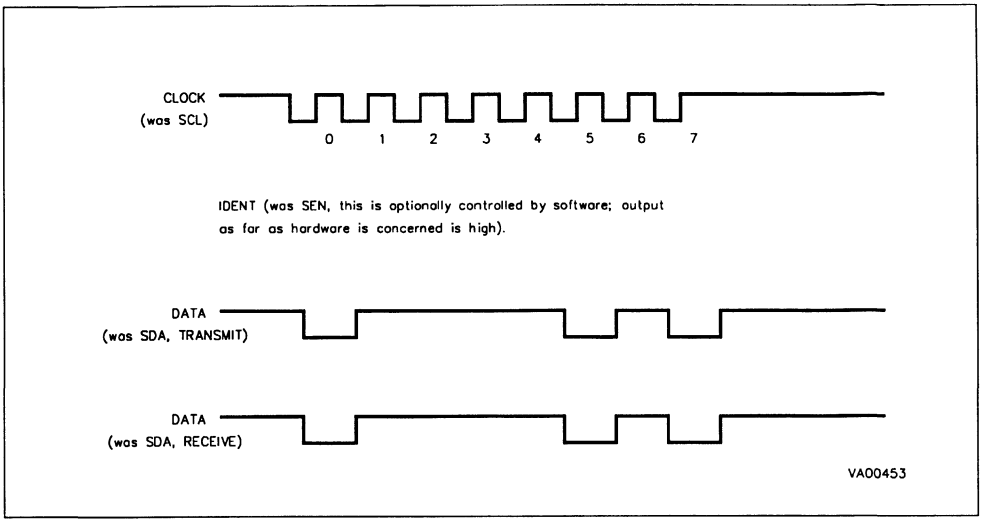
It is also possible to use mixed S-BUS/I²CBUS protocols as showed in Figure 46 (c). S-BUS peripherals will only react to S-BUS protocol signals, while I²CBUS peripherals will only react to I²CBUS signals. Multimaster configuration is not possible with the ST6369 SPI (single master only).

Figure 46. S-BUS/I²C BUS Mixed Configurations



SERIAL PERIPHERAL INTERFACE (Continued)

Figure 47.Sofware Bus (Hardware Bus Disabled) Timing Diagram



STD SPI Protocol (Shift Register)

This protocol is similar to the I^2CBUS with the exception that there is no acknowledge pulse and there are no stop or start bits. The clock cannot be slowed down by the external peripherals.

In this case all three outputs should be high in order not to lock the software I/Os from functioning.

SPI Standard Bus Protocol: The standard bus protocol is selected by loading the SPI Control Register 1 (SCR1 Add. EBH). Bit 0 named I^2C must be set at one and bit 1 named STD mut be reset. When the standard bus protocol is selected bit 2 of the SCR1 is meaningless.

This bit named STOP bit is used only in I^2CBUS or SBUS. However take care thet THE STOP BIT MUST BE RESET WHEN THE STANDARD PROTOCOL IS USED. This bit is set to ZERO after RESET.

14-BIT PWM D/A CONVERTER

The ST6369 PWM D/A CONVERTER (HDA) is composed of a 14-bit counter that allows the conversion of the digital content in an analog voltage, available at the HDA output pin, by using Pulse Width Modification (PWM), and Bit Rate Multiplier (BRM) techniques.

The tuning word consists of a 14-bit word contained in the registers HDADATA1 (location 0EEH) and HDADATA2 (location 0EFH). Coarse tuning (PWM) is performed using the seven MSBits, while fine tuning (BRM) is performed using the data in the seven LSBits. With all zeros loaded the output is zero; as the tuning voltage increases from all zeros, the number of pulses in one period increas to 128 with all pulses being the same width. For values larger than 128, the PWM takes over and the number of pulses in one period remains constant at 128, but the width changes. At the other end of the scale, when almost all ones are loaded, the pulses will start to link together and the number of pulses will decrease. When all ones are loaded, the output will be almost 100% high but will have a low pulse (1/16384 of the high pulse).

Output Details

Inside the on-chip D/A CONVERTER are included the register latches, a reference counter, PWM and BRM control circuitry. In the ST6369 the clock for the 14-bit reference counter is 2MHz derived from the 8MHz system clock. From the circuit point of view, the seven most significant bits control the coarse tuning, while the seven least significant bits control the fine tuning. From the application and software point of view, the 14 bits can be considered as one binary number.

As already mentioned the coarse tuning consists of a PWM signal with 128 steps ; we can consider the fine tuning to cover 128 coarse tuning cycles. The addition of pulses is described in the following Table.

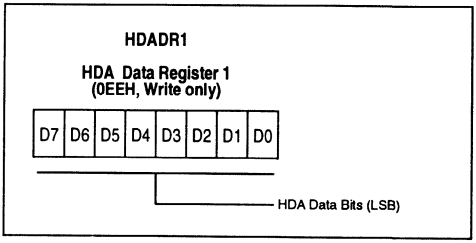
Table 11. Fine Tuning Pulse Addition

Fine Tuning (7 LSB)	N° of Pulses added at the following cycles (0...127)
0000001	64
0000010	32, 96
0000100	16, 48, 80, 112
0001000	8, 24, ...104, 120
0010000	4, 12, ...116, 124
0100000	2, 6, ...122, 126
1000000	1, 3, ...125, 127

The HDA output pin has a standard drive push-pull output configuration.

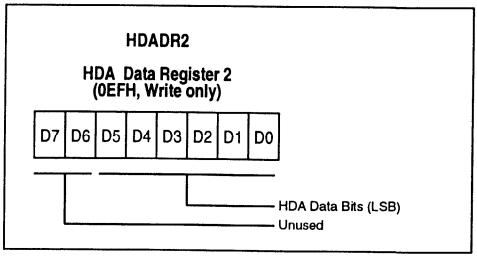
HDA Tuning Cell Registers

Figure 48. HDA Data Register 1



D7-D0. These are the 8 least significant HDA data bits. Bit 0 is the LSB. This register is undefined on reset.

Figure 49. HDA Data Register 2



D7-D6. These bits are not used.

D5-D0. These are the 6 most significant HDA data bits. Bit 5 is the MSB. This register is undefined on reset.

6-BIT PWM D/A CONVERTERS

The D/A macrocell contains up to six PWM D/A outputs (31.25kHz repetition, DA0-DA5) with six bit resolution.

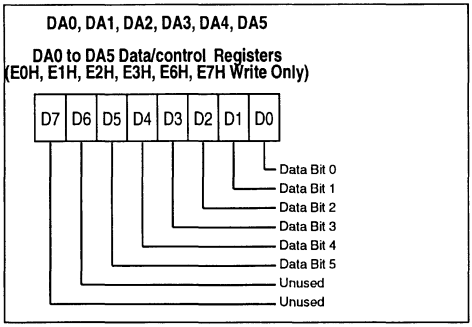
Each D/A converter of ST6369 is composed by the following main blocks:

- pre-divider
- 6-bit counter
- data latches and compare circuits

The pre-divider uses the clock input frequency (8MHz typical) and its output clocks the 6-bit free-running counter. The data latched in the six registers (E0H, E1H, E2H, E3H, E6H and E7H) control the six D/A outputs (DA0,1,2, 3, 4 and 5). When all zeros are loaded the relevant output is an high logic level; all 1's correspond to a pulse with a 1/64 duty cycle and almost 100% zero level.

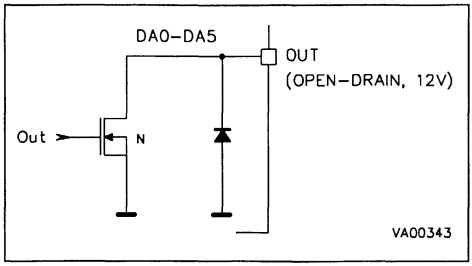
The repetition frequency is 31.25KHz and is related to the 8MHz clock frequency. Use of a different oscillator frequency will result in a different repetition frequency. All D/A outputs are open-drain with standard current drive capability and able to withstand up to 12V.

Figure 50. DA0-DA5 Data/Control Registers



DA0-DA5. These are the 6 bits of the PWM digital to analog converter. Undefined after reset.

Figure 51.6-bit PWM D/A Output Configuration



A/D COMPARATOR

A/D INPUT, HSYNC/PC6 RESULT, VSYNC RESULT AND O0, O1 OUTPUTS

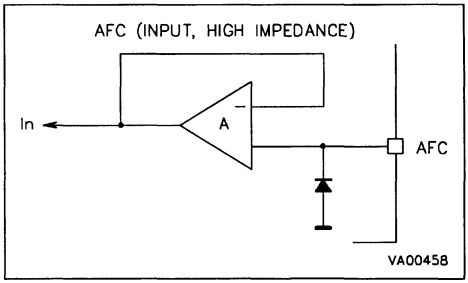
The A/D macrocell contains an A/D comparator with five levels at intervals of 1V from 1V to 5V. The levels can all be lowered by 0.5V to effectively double the resolution.

The A/D used to perform the AFC function (when high threshold is selected) has the following voltage levels: 1,2,3,4 and 5V. Bits 0-2 of AFC result register (E4H address) will provide the result in binary form (less than 1V is 000, greater than 5V is 101).

If the application requires a greater resolution, the sensitivity can be doubled by clearing to zero bit 2 of the OUTPUTS control register, address E5H. In this case all levels are shifted lower by 0.5V. If the two results are now added within a software routine then the A/D S-curve can be located within a resolution of 0.5V.

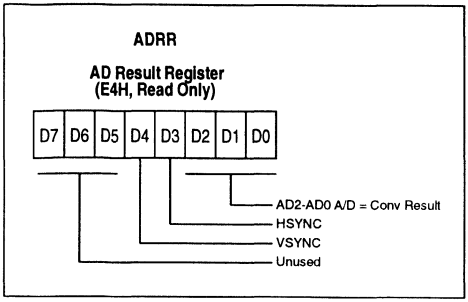
The A/D input has high impedance able to withstand up to 13V signals (input level tolerances ± 200mV absolute and ± 100mv relative to 5V).

Figure 52. A/D Inputs Configuration Diagram



A/D COMPARATOR (Continued)

Figure 53. A/D, HSYNC and VSYNC Result Register



D7-D5. These bits are not used.

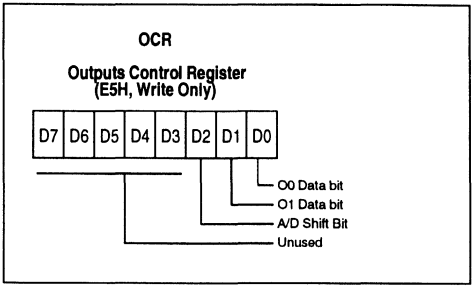
VSYNC. This bit reads the status of the VSYNC pin. It is inverted with respect to the pin.

HSYNC. This bit reads the status of the HSYNC latch. If a signal has been latched this bit will be high.

AD2-AD0. These bits store the real time conversion of the value present on the AD input pin. Undefined reset value.

D7, D6, D5, D4, D3. These bits are not used.

Figure 54. Outputs Control Register



A/D Shift. This bit determines the voltage range of the AFC input. Writing a zero will select the 0.5V to 4.5V range. Writing a one will select the 1.0V to 5.0V range. Undefined after reset.

O1,O0. These bits control the output pins O1,O0. They are undefined after reset.

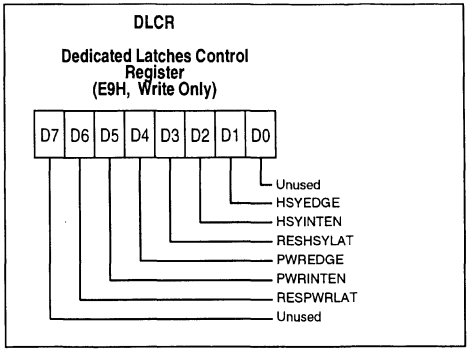
DEDICATED LATCHES

Two latches are available which may generate interrupts to the ST6369 core.

The HSYNC latch is set either by the falling or rising edge of the signal on pin PC6(HSYNC). If bit 1 (HSYEDGE) of the latches register (E9H) is high, then the latch will be triggered on the rising edge of the signal at PC6(HSYNC). If bit 1 (HSYEDGE) is low, then the latch will be triggered on the falling edge of the signal at PC6(HSYNC). The HSYNC latch can be reset by setting bit 3 (RESHSYLAT) of the latches register; the bit is set only and a high should be written every time the HSYNC latch needs to be reset. If bit 2 (HSYINTEN) of the latches register (E9H) is high, then the output of the HSYNC latch, HSYNCN, may generate an interrupt (#0). HSYNCN is inverted with respect to the state of the HSYNC latch. If bit 2 (HSYINTEN) is low, then the output of the HSYNC latch, HSYNCN, is forced high. The state of the HSYNC latch may be read from bit 3 (HSYNC) of register E4H; if the HSYNC latch is set, then bit 3 will be high.

The PWR latch is set either by the falling or rising edge of the signal on pin PC4(PWRIN). If bit 4 (PWREDGE) of the latches register (E9H) is high, then the latch will be triggered on the rising edge of the signal at PC4(PWRIN). If bit 4 (PWREDGE) is low, then the latch will be triggered on the falling edge of the signal at PC4(PWRIN). The PWR latch can be reset by setting bit 6 (RESPWRLAT) of the latches register; the bit is set only and a high should be written every time the PWR latch needs to be reset. If bit 5 (PWRINTEN) of the latches register (E9H) is high, then the output of the PWR latch, PWRINTN, may generate an interrupt (#4). PWRINTN is inverted with respect to the state of the PWR latch. If bit 5 (PWRINTEN) is low, then the output of the PWR latch, PWRINTN, is forced high.

Figure 55. Dedicated Latches Control Register



D0. This bit is not used

D7. This bit is not used

RESPWRLAT. Resets the PWR latch; this bit is set only.

PWRINTEN. This bit enables the PWRINTN signal (#4) from the latch to the ST6369 core. Undefined after reset.

PWREDGE. The bit determines the edge which will cause the PWRIN latch to be set. If this bit is high, than the PWRIN latch will be set on the rising edge of the PWRIN signal. Undefined after reset.

RESHSYLAT. Resets the HSYNC latch; this bit is set only.

HSYINTEN. This bit enables the HSYNCN signal (#0) from the latch to the ST6369 core. Undefined after reset.

HSYEDGE. The bit determines the edge which will cause the HSYNC latch to be set. If this bit is high, than the HSYNC latch will be set on the rising edge of the HSYNC signal. Undefined after reset.

SOFTWARE DESCRIPTION

The ST6369 software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum; in short to provide byte efficient programming capability. The ST6369 Core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

Addressing Modes

The ST6369 Core has 9 addressing modes which are described in the following paragraphs. The ST6369 Core uses three different address spaces: Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte that is processed by the instruction is stored in the location that follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The Core can address the four RAM registers X,Y,V,W (locations 80H, 81H, 82H, 83H) in the short-direct addressing mode. In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80H and 81H are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) that use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction that follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits that characterize the kind of the test, one bit that determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits that give the span of the branch (0H to FH) that must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80H,81H). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

SOFTWARE DESCRIPTION (Continued)

Instruction Set

The ST6369 Core has a set of 40 basic instructions. When these instructions are combined with nine addressing modes, 244 usable opcodes can be obtained. They can be divided into six different types:load/store, arithmetic/logic, conditional branch, control instructions, jump/call, bit manipulation. The following paragraphs describe the different types.

All the instructions within a given type are presented in individual tables.

Load & Store. These instructions use one,two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data. See Table 12.

SOFTWARE DESCRIPTION (Continued)

Table 12. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
#. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
*. Not Affected

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator. See Table 13.

SOFTWARE DESCRIPTION (Continued)

Table 13. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	*
AND A, (Y)	Indirect	1	4	Δ	*
AND A, rr	Direct	2	4	Δ	*
ANDI A, #N	Immediate	2	4	Δ	*
CLR A	Short Direct	2	4	Δ	Δ
CLR rr	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
#. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
*. Not Affected

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met. See Table 14.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations. See Table 15.

Control Instructions. The control instructions control the MCU operations during program execution. See Table 16-

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space. Refer to Table 17.

Table 14. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ. Affected
*. Not Affected

Table 15. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
*. Not Affected

Table 16. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated and a WAIT is automatically executed instead of a STOP if the hardware activated watchdog function is selected.
Δ. Affected
*. Not Affected

Table 17. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc.12-bit address;
*. Not Affected

SOFTWARE DESCRIPTION (Continued)

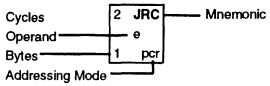
Opcode Map Summary. The following table contains an opcode map for the instructions used on the MCU.

Low / Hi	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Low / Hi
0 0000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 LD a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 pcr	4 LD a,(y) 1 ind	0 0000
1 0001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC x 1 sd	2 JRC e 1 prc	4 LDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 JRZ e 1 pcr	4 DEC x 1 sd	2 JRC e 1 pcr	4 LD a,rr 2 dir	1 0001
2 0010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b4,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 CP a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b4,rr 2 b.d	2 JRZ e 1 pcr	4 COM a 1 inh	2 JRC e 1 pcr	4 CP a,(y) 1 ind	2 0010
3 0011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b4,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,x 1 sd	2 JRC e 1 prc	4 CPI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b4,rr 2 b.d.	2 JRZ e 1 pcr	4 LD x,a 1 sd	2 JRC e 1 pcr	4 CP a,rr 2 dir	3 0011
4 0100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b2,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 ADD a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b2,rr 2 b.d	2 JRZ e 1 pcr	2 RETI 1 inh	2 JRC e 1 pcr	4 ADD a,(y) 1 ind	4 0100
5 0101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b2,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC y 1 sd	2 JRC e 1 prc	4 ADDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b2,rr 2 b.d	2 JRZ e 1 pcr	4 DEC y 1 sd	2 JRC e 1 pcr	4 ADD a,rr 2 dir	5 0101
6 0110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b6,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 INC (x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b6,rr 2 b.d	2 JRZ e 1 pcr	2 STOP 1 inh	2 JRC e 1 pcr	4 INC (y) 1 ind	6 0110
7 0111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b6,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,y 1 sd	2 JRC e 1 prc	# 	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b6,rr 2 b.d	2 JRZ e 1 pcr	4 LD y,a 1 sd	2 JRC e 1 pcr	4 INC rr 2 dir	7 0111
8 1000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b1,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 LD (x),a 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b1,rr 2 b.d	2 JRZ e 1 pcr	# 	2 JRC e 1 pcr	4 LD (y),a 1 ind	8 1000
9 1001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b1,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC v 1 sd	2 JRC e 1 prc	# 	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b1,rr 2 b.d	2 JRZ e 1 pcr	4 DEC v 1 sd	2 JRC e 1 pcr	4 LD rr,a 2 dir	9 1001
A 1010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b5,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 AND a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b5,rr 2 b.d	2 JRZ e 1 pcr	4 RLC a 1 inh	2 JRC e 1 pcr	4 AND a,(y) 1 ind	A 1010
B 1011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b5,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,v 1 sd	2 JRC e 1 prc	4 ANDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b5,rr 2 b.d	2 JRZ e 1 pcr	4 LD v,a 1 sd	2 JRC e 1 pcr	4 AND a,rr 2 dir	B 1011
C 1100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b3,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 SUB a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b3,rr 2 b.d	2 JRZ e 1 pcr	2 RET 1 inh	2 JRC e 1 pcr	4 SUB a,(y) 1 ind	C 1100
D 1101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b3,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC w 1 sd	2 JRC e 1 prc	4 SUBI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b3,rr 2 b.d	2 JRZ e 1 pcr	4 DEC w 1 sd	2 JRC e 1 pcr	4 SUB a,rr 2 dir	D 1101
E 1110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b7,rr,ee 3 bt	2 JRZ e 1 pcr	# 	2 JRC e 1 prc	4 DEC (x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b7,rr 2 b.d	2 JRZ e 1 pcr	2 WAIT 1 inh	2 JRC e 1 pcr	4 DEC (y) 1 ind	E 1110
F 1111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b7,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,w 1 sd	2 JRC e 1 prc	# 	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b7,rr 2 b.d	2 JRZ e 1 pcr	4 LD w,a 1 sd	2 JRC e 1 pcr	4 DEC rr 2 dir	F 1111

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:
# Indicates Illegal Instructions
e 5 Bit Displacement
b 3 Bit Address
rr1byte dataspace address
nn 1 byte immediate data
abc 12 bit address
ee 8 Bit Displacement



ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from :

$$Tj = T_A + PD \times RthJA$$

Where : T_A = Ambient Temperature.

RthJA = Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint = $I_{DD} \times V_{DD}$ (chip internal power).

Pport = Port power dissipation (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage (AD IN)	V_{SS} - 0.3 to +13	V
V_I	Input Voltage (Other Inputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
V_O	Output Voltage (PA4-PA7, PC4-PC7, DA0-DA5)	V_{SS} - 0.3 to +13	V
V_O	Output Voltage (Other Outputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
I_O	Current Drain per Pin Excluding V_{DD}, V_{SS}, PA6, PA7	± 10	mA
I_O	Current Drain per Pin (PA6, PA7)	± 50	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	150	mA
Tj	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Note : Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device . This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PSDIP42			67	°C/W

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	1 Suffix Version	0		70	°C
V_{DD}	Operating Supply Voltage		4.5	5.0	6.0	V
f_{OSC}	Oscillator Frequency RUN & WAIT Modes			8	8.1	MHz
f_{OSDOSC}	On-screen Display Oscillator Frequency				8.0	MHz

EEPROM INFORMATION

The ST63xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = 0 to +70°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	All I/O Pins			0.2xV_{DD}	V
V_{IH}	Input High Level Voltage	All I/O Pins	0.8xV_{DD}			V
V_{HYS}	Hysteresis Voltage(1)	All I/O Pins V_{DD} = 5V		1.0		V
V_{OL}	Low Level Output Voltage	DA0-DA5, PB1-PB2, PB4-PB6, PC0-PC7, O0, O1, PA0-PA5 V_{DD} = 4.5V I_{OL} = 1.6mA I_{OL} = 5.0mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	PA6-PA7 V_{DD} = 4.5V I_{OL}= 1.6mA I_{OL}= 25mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	OSCOUT V_{DD} = 4.5V I_{OL}= 0.4mA			0.4	V
V_{OL}	Low Level Output Voltage	HDA Output V_{DD} = 4.5V I_{OL}= 0.5mA I_{OL}= 1.6mA			 0.4 1.0	 V V
V_{OH}	High Level Output Voltage	PB1-PB2, PB4-PB6, PA0-PA3, V_{DD} = 4.5V I_{OH} = – 1.6mA	4.1			V
V_{OH}	High Level Output Voltage	OSCOUT, V_{DD} = 4.5V I_{OH}= – 0.4mA	4.1			V
V_{OH}	High Level Output Voltage	HDA Output V_{DD} = 4.5V I_{OH}= - 0.5mA	4.1			V
I_{PU}	Input Pull Up Current Input Mode with Pull-up	PB1-PB2, PB4-PB6, PA0-PA3, PC0-PC3 V_{IN}= V_{SS}	– 100	– 50	– 25	mA
I_{IL} I_{IH}	Input Leakage Current	OSCIN V_{IN}= V_{SS} V_{IN}= V_{DD}	– 10 0.1	– 1 1	– 0.1 10	µA
I_{IL}	Input Pull-down current in Reset	OSCIN	100			µA
I_{IL} I_{IH}	Input Leakage Current	All I/O Input Mode no Pull-up V_{IN}= V_{DD} or V_{SS}	– 10		10	µA

DC ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
$V_{DD}RAM$	RAM Retention Voltage in RESET		1.5			V
I_{IL} I_{IH}	Input Leakage Current	Reset Pin with Pull-up $V_{IN}= V_{SS}$	– 50	– 30	– 10	µA
I_{IL} I_{IH}	Input Leakage Current	AD Pin $V_{IH}= V_{DD}$ $V_{IL}= V_{SS}$ $V_{IH}= 12.0V$	– 1		1 40	µA
I_{OH}	Output Leakage Current	DA0-DA5, PA4-PA5, PC0-PC7, O0, O1 $V_{OH} = V_{DD}$			10	µA
I_{OH}	Output Leakage Current High Voltage	DA0-DA5, PA4-PA7, PC4-PC7, O0, O1 $V_{OH} = 12V$			40	µA
I_{DD}	Supply Current RUN Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6.0V		6	16	mA
I_{DD}	Supply Current WAIT Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6V		3	10	mA
I_{DD}	Supply Current at transition to RESET	f_{OSC}= Not App, ILoad= 0mA V_{DD}= 6V		0.1	1	mA
V_{ON}	Reset Trigger Level ON	RESET Pin			$0.3xV_{DD}$	V
V_{OFF}	Reset Trigger Level OFF	RESET Pin	$0.8xV_{DD}$			V
V_{TA}	Input Level Absolute Tolerance	AD Pin $V_{DD} = 5V$			±200	mV
V_{TR}	Input Level Relatice Tolerance (1)	AD Pin Relative to other levels $V_{DD} = 5V$			±100	mV

Note: 1. Not 100% Tested

AC ELECTRICAL CHARACTERISTICS

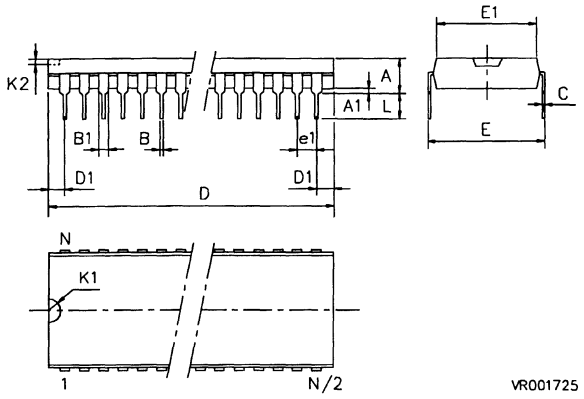
(T_A = 0 to +70°C, f_{OSC}=8MHz, V_{DD}=4.5 to 6.0V unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{WRES}	Minimum Pulse Width	RESET Pin	125			ns
tO_{HL}	High to Low Transition Time	PA6, PA7 V_{DD} = 5V, CL = 1000pF (2)		100		ns
tO_{HL}	High to Low Transition Time	DA0-DA5, PB1-PB2, PB4-PB6, PC0-PC7, V_{DD} = 5V, CL = 100pF		20		ns
tO_{LH}	Low to High Transition Time	PA0-PA3, PB1-PB2, PB4-PB6, PC0-PC3 V_{DD} = 5V, CL = 100pF		20		ns
tO_H	Data HOLD Time SPI after clock goes low I^2CBUS/S-BUS Only		175			ns
f_{DA}	D/A Converter Repetition Frequency(1)			31.25		kHz
f_{SIO}	SIO Baud Rate(1)			62.50		kHz
t_{WEE}	EEPROM Write Time	T_A = 25°C One Byte		5	10	ms
Endurance	EEPROM WRITE/ERASE Cycles	Q_A LOT Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention (4)	T_A = 25°C	10			years
C_{IN}	Input Capacitance (3)	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance (3)	All outputs Pins			10	pF
COSCIN, COSCOUT	Oscillator Pins Internal Capacitance(3)			5		pF

Notes:
1. A clock other than 8 MHz will affect the frequency response of those peripherals (D/A, 62.5kHz and SPI) whose clock is derived from the system clock.
2. The rise and fall times of PORT A have been reduced in order to avoid current spikes while maintaining a high drive capability
3. Not 100% Tested
4. Based on extrapolated data

PACKAGE MECHANICAL DATA

Figure 56. ST6369 40 Pin Plastic Dual-In-Line Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.2		4.8	0.086		0.189
A1	0.51		1.77	0.010		0.069
B	0.38		0.58	0.015		0.023
B1	0.97		1.52	0.055		0.065
C	0.2		0.3	0.008		0.009
D	50.30		52.22	1.980		20.560
D1	–	–	–	–	–	–
E			16.3			0.641
E1		12.9			0.508	
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	3.18		4.44	1.25		0.174
e1		2.54			0.10	
Number of Pins						
N	40					

ORDERING INFORMATION

The following chapter deals with the procedure for transfer the Program/Data ROM codes to SGS-THOMSON.

Communication of the ROM Codes. To communicate the contents of Program /Data ROM memories to SGS-THOMSON, the customer has to send:

- one file in INTEL INTELLEC 8/MDS FORMAT (either as an EPROM or in a MS-DOS 5" diskette) for the PROGRAM Memory
- one file in INTEL INTELLEC 8/MDS FORMAT (either as an EPROM or in a MS-DOS 5" diskette) for the EEPROM initial content (this file is optional)

The program ROM should respect the ROM Memory Map as in Table 18.

The ROM code must be generated with ST6 assembler. Before programming the EPROM, the buffer of the EPROM programmer must be filled with FFH.

For shipment to SGS-THOMSON the EPROMs should be placed in a conductive IC carrier and packaging carefully.

Customer EEPROM Initial Contents: Format

a. The content should be written into an INTEL INTELLEC format file.

b. In the case of 384 bytes of EEPROM, the starting address is 000H and the end address is 7FH. The order of the pages (64 bytes each) is an in the specification (ie. b7, b1 b0: 001, 010, 011, 101, 110. 111).

c. Undefined or don't care bytes should have the content FFH.

Listing Generation & Verification. When SGS-THOMSON receives the Codes, they are compared and a computer listing is generated from them. This listing refers extractly to the mask that will be used to produce the microcontroller. Then the listing is returned to the customer that must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed list constitutes a part of the contractual agreement for the creation of the customer mask. SGS-THOMSON sales organization will provide detailed information on contractual points.

ST6369 MICROCONTROLLER OPTION LIST

Table 18. ROM Memory Map

ROM Page	Device Address	EPROM Address (1)	Description
Page 0	0000H-007FH 0080H-07FFH	0000H-007FH 0080H-07FFH	Reserved User ROM
Page 1 "STATIC"	0800H-0F9FH 0FA0H-0FEFH 0FF0H-0FF7H 0FF8H-0FFBH 0FFCH-0FFDH 0FFEH-0FFFH	0800H-0F9FH 0FA0H-0FEFH 0FF0H-0FF7H 0FF8H-0FFBH 0FFCH-0FFDH 0FFEH-0FFFH	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
Page 2	0000H-000FH 0010H-07FFH	1000H-100FH 1010H-17FFH	Reserved User ROM
PAGE 3	0000H-000FH 0010H-07FFH	1800H-180FH 1810H-1FFFH	Reserved user ROM

Notes:

1. EPROM addresses are related to the use of ST63E69 emulation devices.

ORDERING INFORMATION TABLE

Sales Type	ROM/EEPROM Size	D/A Converter	Temperature Range	Package
ST6369B1/XX	8K/384 Bytes	7	0 to + 70 ° C	PDIP40

Note: "XX" Is the ROM code identifier that is allocated by SGS-THOMSON after receipt of all required options and the related ROM file.

Customer: . . . . . .
Address: . . . . . .
Contact: . . . . . .
Phone No: . . . . . .
Reference: . . . . . .

Package [ ] (p) Temperature Range [ ] (t)
For marking one line with 16 characters maximum is possible

Special Marking [ ] (y/n) Line1 "_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _" (N)

Notes:
(p) B= Dual in Line Plastic
(t) 1= 0 to 70°C

(N) Letters, digits, ' . ', ' - ', ' / ' and spaces only

Marking: the default marking is equivalent to the sales type only (part number).

CHECK LIST:

	YES	NO
ROM CODE	[ ]	[ ]
EEPROM Code (if Desired)	[ ]	[ ]

Signature

Date

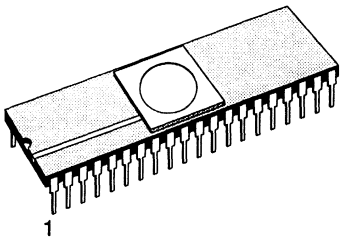
SGS-THOMSON MICROELECTRONICS

ST63E69
ST63T69

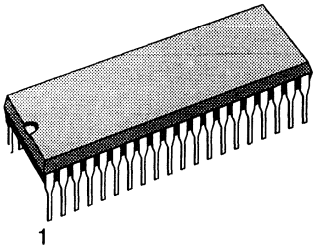
8-BIT EPROM HCMOS MCUs FOR DIGITAL CONTROLLED MULTI FREQUENCY MONITOR

PRELIMINARY DATA

- 4.5 to 6V supply operating range
- 8MHz Maximum Clock Frequency
- User Program EPROM: 7948 bytes
- Reserved Test EPROM: 244 bytes
- Data ROM: user selectable size
- Data RAM: 256 bytes
- Data EEPROM: 384 bytes
- 40-Pin Ceramic Dual in Line Package for EPROM version
- 40-Pin Plastic Dual in Line Package for OTP version
- Up to 23 software programmable general purpose Inputs/Outputs, including 2 direct LED driving Outputs
- Two Timers each including an 8-bit counter with a 7-bit programmable prescaler
- Digital Watchdog Function
- Serial Peripheral Interface (SPI) supporting S-BUS/ I^2C BUS and standard serial protocols
- One 14-Bit PWM D/A Converter
- Six 6-Bit PWM D/A Converters
- One A/D converter with 0.5V resolution
- Five interrupt vectors (HSYNC/NMI, Timer 1 & 2, VSYNC, PWR INT.)
- On-chip clock oscillator
- These EPROM and OTP versions are fully pin to pin compatible with ST6369 ROM version.
- The development tool of the ST6369 microcontrollers consists of the ST6369-EMU emulation and development system to be connected via a standard RS232 serial line to an MS-DOS Personal Computer.
- EPROM programming board ST6369-EPB



CDIP40W



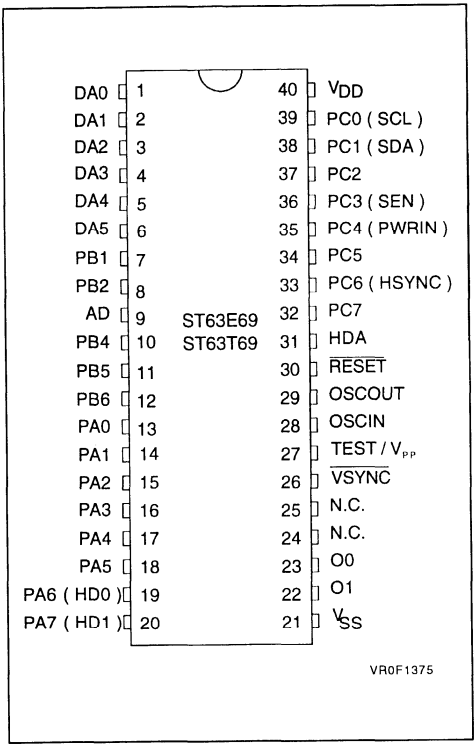
PDIP40

(Ordering Information at the end of the datasheet)

DEVICE SUMMARY

EPROM DEVICE	OTP DEVICE	EPROM (Bytes)	EEPROM (Bytes)	D/A Conv.
ST63E69	ST63T69	8K	384	7

Figure 1. ST63E69, T69 Pin Configuration



GENERAL DESCRIPTION

The ST63E69 microcontroller is member of the 8-bit HCMOS ST638x family, a series of devices specially oriented to Digital Controlled Multi Frequency Monitor applications. They are the EPROM/OTP versions of the ST6369 ROM device and are suitable for product prototyping and low volume production. ST6369 is based on a building block approach: a common core is surrounded by a combination of on-chip peripherals (macrocells) available from a standard library. These peripherals are designed with the same Core technology providing full compatibility and short design time. Many of these macrocells are specially dedicated to DCMF monitor applications. The macrocells of the ST6369 are: two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), a digital hardware activated watchdog function (DHWD), a 14-bit voltage synthesis tuning peripheral, a Serial Peripheral Interface (SPI), six 6-bit PWM D/A converters, an A/D converter with 0.5V resolution, a 14-bit PWM D/A converter. In addition the following memory resources are available: program EPROM (8K), data RAM (256 bytes), EEPROM (384 bytes).

Figure 2. ST63E69, T69 Block Diagram

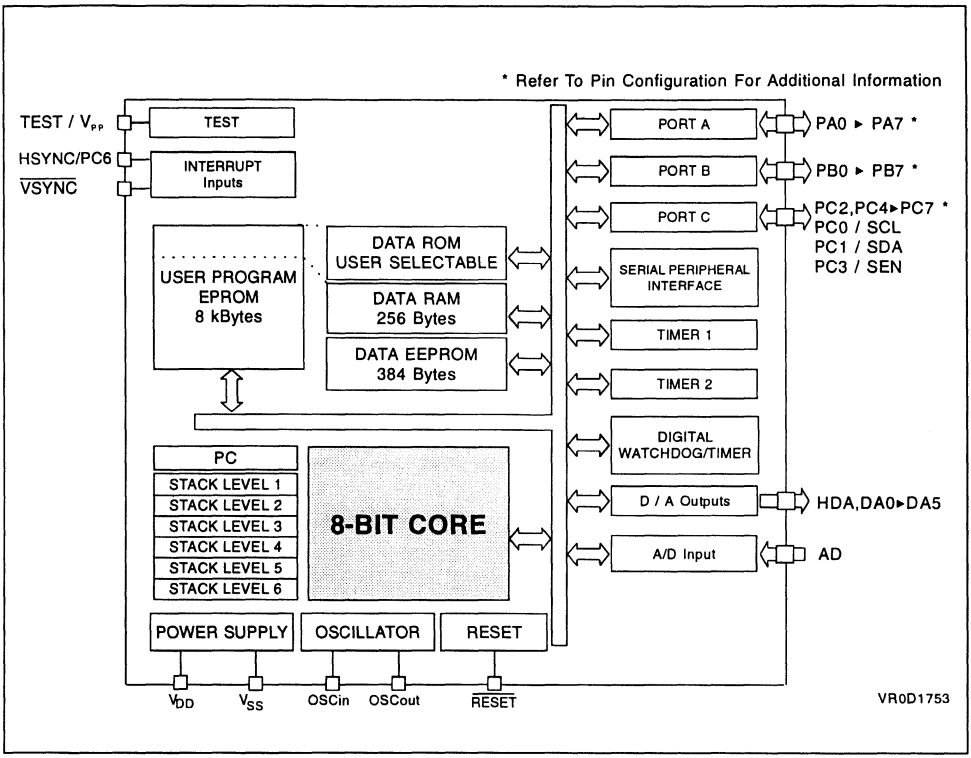


Table 1. Device Summary

DEVICE	EPROM (Bytes)	OTPROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	A/D	14-bit D/A	6-bit D/A	TARGET ROM DEVICE
ST63E69	8K		256	384	1	1	6	ST6369
ST63T69		8K	256	384	1	1	6	ST6369

PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCIN, OSCOUT. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. The OSCIN pin is the input pin, the OSCOUT pin is the output pin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to start the microcontroller to the beginning of its program. Additionally the quartz crystal oscillator will be disabled when the $\overline{\text{RESET}}$ pin is low to reduce power consumption during reset phase.

TEST/V_{PP}. The TEST pin must be held at V_{SS} for normal operation.

If this pin is connected to a +12.5V level during the reset phase, The EPROM programming mode is entered.

CAUTION: *Exceeding 13V on TEST/V_{PP} pin will permanently damaged the device*

PA0-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured as either an input with or without pull-up resistor or as an output under software control of the data direction register. Pins PA4 to PA7 are configured as open-drain outputs (12V drive). On PA4-PA7 pins the input pull-up option is not available while PA6 and PA7 have additional current driving capability (25mA, V_{OL}:1V). PA0 to PA3 pins are configured as push-pull.

PB1-PB2, PB4-PB6. These 5 lines are organized as one I/O port (B). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register.

PC0-PC7. These 8 lines are organized as one I/O port (C). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register. Pins PC0 to PC3 are configured as open-drain (5V drive) in output mode while PC4 to PC7 are open-drain with 12V drive and the input pull-up options does not exist on these four pins. PC0, PC1 and PC3 lines when in output mode are "ANDed" with the SPI control signals and are all open-drain. PC0 is connected to the SPI clock signal (SCL), PC1 with the SPI data signal (SDA) while PC3 is connected with SPI enable signal (SEN, used in S-BUS protocol). Pin PC4 and PC6 can also be inputs to software programmable edge sensitive latches which can generate interrupts; PC4 can be connected to Power Interrupt while PC6 can be connected to the HSYNC/NMI interrupt line.

DA0-DA5. These pins are the six PWM D/A outputs of the 6-bit on-chip D/A converters. These lines have open-drain outputs with 12V drive. The output repetition rate is 31.25KHz (with 8MHz clock).

AD. This is the input of the on-chip 10 levels comparator that can be used to implement the Analog Keyboard function. This pin is an high impedance input able to withstand signals with a peak amplitude up to 12V.

VSYNC. This is the Vertical Synchronization pin. This pin is connected to an internal timer interrupt.

O0, O1. These two lines are output open-drain pins with 12V drive.

HDA. This is the output pin of the on-chip 14-bit PWM D/A Converter. This line is a push-pull output with standard drive.

Table 2. Pin Summary

Pin Function	Description
DA0 to DA5	Output, Open-Drain, 12V
AD	Input, High Impedance, 12V
HDA	Output, Push-Pull
TEST/V_{PP}	Input, Pull-Down, V_{PP} EPROM Programming Voltage Input
OSCIN	Input, Resistive Bias, Schmitt Trigger to Reset Logic Only
OSCOUT	Output, Push-Pull
RESET	Input, Pull-up, Schmitt Trigger Input
PA0-PA3	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PA4-PA5	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
PA6-PA7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input, High Drive
PB1-PB2	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PB4-PB6	I/O, Push-Pull, Software Input Pull-up, Schmitt Trigger Input
PC0-PC3	I/O, Open-Drain, 5V , Software Input Pull-up, Schmitt Trigger Input
PC4-PC7	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger Input
O0, O1	Output, Open-Drain, 12V
V_{DD}, V_{SS}	Power Supply Pins

ST63E69,T69 EPROM/OTP DESCRIPTION.

The ST63E69 is the EPROM version of the ST6369 ROM product. It is intended for use during the development of an application, and for pre-production and small volume production. The ST63T69 OTP has the same characteristics. They both include EPROM memory instead of the ROM memory of the ST6369, and so the program and constants of the program can be easily modified by the user with the ST63E69 EPROM programming board from SGS-THOMSON.

The Table 3 is a summary of the EPROM/ROM map and its reserved area.

From a user point of view (with the following exceptions) the ST63E69,T69 products have exactly the same software and hardware features of the ROM version. An additional mode is used to configure the part for programming of the EPROM, this is set by a +12.5V voltage applied to the TEST/V_{PP} pin. The programming of the ST63E69,T69 is described in the User Manual of the EPROM Programming board.

On the ST63E69, all the 7948 bytes of PROGRAM memory are available for the user, as all the EPROM memory can be erased by exposure to UV light. On the ST63T69 (OTP device) a reserved area for test purposes exists, as for the ST6369 ROM device. In order to avoid any discrepancy between program functionality when using the EPROM, OTP and ROM it is recommended not to use these reserved areas, even when using the ST63E69.

THE READER IS ASKED TO REFER TO THE DATASHEET OF THE ST6369 ROM-BASED DEVICE FOR FURTHER DETAILS.

EPROM ERASING

The EPROM of the windowed package of the ST63E69 may be erased by exposure to Ultra Violet light.

The erasure characteristic of the ST63E69 EPROM is such that erasure begins when the memory is exposed to light with wave lengths shorter than approximately 4000Å. It should be noted that sunlight and some types of fluorescent lamps have wavelengths in the range 3000-4000Å. It is thus recommended that the window of the ST63E69 package be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.
The recommended erasure procedure of the ST63E69 EPROM is exposure to short wave ultraviolet light which has wavelength 2537Å. The integrated dose (i.e. UV intensity x exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm^2 power rating. The ST63E69 should be placed within 2.5 cm (1 inch) of the lamp tubes during erasure.

Table 3. EPROM/ROM Map

ROM Page	Device Address	Description
PAGE 0	0000H-007FH 0080H-07FFH	Reserved User ROM
PAGE 1 "STATIC"	0800H-0F9FH 0FA0H-0FEFH 0FF0H-0FF7H 0FF8H-0FFBH 0FFCH-0FFDH 0FFEH-0FFFH	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
PAGE 2	0000H-000FH 0010H-07FFH	Reserved User ROM
PAGE 3	0000H-000FH 0010H-07FFH	Reserved User ROM

ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from :

$Tj = T_A + P_D \times RthJA$

Where : T_A = Ambient Temperature.

RthJA= Package thermal resistance (junction-to ambient).

$P_D = Pint + Pport$.

Pint = $I_{DD} \times V_{DD}$ (chip internal power).

Pport = Port power dissipation (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage (AD IN)	V_{SS} - 0.3 to +13	V
V_I	Input Voltage (Other Inputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
V_O	Output Voltage (PA4-PA7, PC4-PC7, DA0-DA5)	V_{SS} - 0.3 to +13	V
V_O	Output Voltage (Other Outputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
V_{PP}	EPROM programming Voltage	-0.3 to 13.0	V
I_O	Current Drain per Pin Excluding V_{DD}, V_{SS}, PA6, PA7	± 10	mA
I_O	Current Drain per Pin (PA6, PA7)	± 50	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	150	mA
Tj	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Note : Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device . This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature		0		70	°C
V_{DD}	Operating Supply Voltage		4.5	5.0	6.0	V
V_{PP}	EPROM programming Voltage		12.0	12.5	13.0	V
f_{OSC}	Oscillator Frequency RUN & WAIT Modes			8	8.1	MHz

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PDIP40			67	°C/W

EEPROM INFORMATION

The ST63xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = 0 to +70°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	All I/O Pins			$0.2xV_{DD}$	V
V_{IH}	Input High Level Voltage	All I/O Pins	$0.8xV_{DD}$			V
V_{HYS}	Hysteresis Voltage(1)	All I/O Pins V_{DD} = 5V		1.0		V
V_{OL}	Low Level Output Voltage	DA0-DA5, PB1-PB2, PB3-PB6 PC0-PC7, O0, O1, PA0-PA5 V_{DD} = 4.5V I_{OL} = 1.6mA I_{OL} = 5.0mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	PA6-PA7 V_{DD} = 4.5V I_{OL}= 1.6mA I_{OL}= 25mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	OSCOUT V_{DD} = 4.5V I_{OL}= 0.4mA			 0.4	 V
V_{OL}	Low Level Output Voltage	HDA Output V_{DD} = 4.5V I_{OL}= 0.5mA I_{OL}= 1.6mA			 0.4 1.0	 V V
V_{OH}	High Level Output Voltage	PB1-PB2, PB3-PB6, PA0-PA3 V_{DD} = 4.5V I_{OH} = – 1.6mA	4.1			V
V_{OH}	High Level Output Voltage	OSCOUT, V_{DD} = 4.5V I_{OH}= – 0.4mA	4.1			V
V_{OH}	High Level Output Voltage	HDA Output V_{DD} = 4.5V I_{OH}= - 0.5mA	4.1			V

DC ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
I_{PU}	Input Pull Up Current Input Mode with Pull-up	PB1-PB2, PB3-PB6, PA0-PA3, PC0-PC3 V_{IN}= V_{SS}	– 100	– 50	– 25	mA
I_{IL} I_{IH}	Input Leakage Current	OSCIN V_{IN}= V_{SS} V_{IN}= V_{DD}	– 10 0.1	– 1 1	– 0.1 10	µA
I_{IL}	Input Pull-down current in Reset	OSCIN	100			µA
I_{IL} I_{IH}	Input Leakage Current	All I/O Input Mode no Pull-up V_{IN}= V_{DD} or V_{SS}	– 10		10	µA
V_{DD}RAM	RAM Retention Voltage in RESET		1.5			V
I_{IL} I_{IH}	Input Leakage Current	Reset Pin with Pull-up V_{IN}= V_{SS}	– 50	– 30	– 10	µA
I_{IL} I_{IH}	Input Leakage Current	AD Pin V_{IH}= V_{DD} V_{IL}= V_{SS} V_{IH}= 12.0V	– 1		1 40	µA
I_{OH}	Output Leakage Current	DA0-DA5, PA4-PA5, PC0-PC7, O0, O1 V_{OH} = V_{DD}			10	µA
I_{OH}	Output Leakage Current High Voltage	DA0-DA5, PA4-PA7, PC4-PC7, O0, O1 V_{OH} = 12V			40	µA
I_{DD}	Supply Current RUN Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6.0V		6	16	mA
I_{DD}	Supply Current WAIT Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6V		3	10	mA
I_{DD}	Supply Current at transition to RESET	f_{OSC}= Not App, ILoad= 0mA V_{DD}= 6V		0.1	1	mA
V_{ON}	Reset Trigger Level ON	RESET Pin			0.3xV_{DD}	V
V_{OFF}	Reset Trigger Level OFF	RESET Pin	0.8xV_{DD}			V
V_{TA}	Input Level Absolute Tolerance	AD Pin V_{DD} = 5V			±200	mV
V_{TR}	Input Level Relatice Tolerance (1)	AD Pin Relative to other levels V_{DD} = 5V			±100	mV

Note: 1. Not 100% Tested

AC ELECTRICAL CHARACTERISTICS

(T_A = 0 to +70°C, f_{OSC}=8MHz, V_{DD}=4.5 to 6.0V unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{WRES}	Minimum Pulse Width	RESET Pin	125			ns
tO_{HL}	High to Low Transition Time	PA6, PA7 V_{DD} = 5V, CL = 1000pF (2)		100		ns
tO_{HL}	High to Low Transition Time	DA0-DA5, PB1-PB2, PB4-PB6 PC0-PC7 V_{DD} = 5V, CL = 100pF		20		ns
tO_{LH}	Low to High Transition Time	PB1-PB2 , PB4-PB6, PA0-PA3, PC0-PC3 V_{DD} = 5V, CL = 100pF		20		ns
tO_H	Data HOLD Time SPI after clock goes low I^2CBUS/S-BUS Only		175			ns
f_{DA}	D/A Converter Repetition Frequency(1)			31.25		kHz
f_{SIO}	SIO Baud Rate(1)			62.50		kHz
t_{WEE}	EEPROM Write Time	T_A = 25°C, One Byte		5	10	ms
Endurance	EEPROM WRITE/ERASE Cycles	Q_A L_{OT} Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention (4)	T_A = 25°C	10			years
C_{IN}	Input Capacitance (3)	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance (3)	All outputs Pins			10	pF
COSCIN, COSCOUT	Oscillator Pins Internal Capacitance(3)			5		pF

Notes:
1. A clock other than 8 MHz will affect the frequency response of those peripherals (D/A, 62.5kHz and SPI) whose clock is derived from the system clock.
2. The rise and fall times of PORT A have been reduced in order to avoid current spikes while maintaining a high drive capability
3. Not 100% Tested
4. Based on extrapolated data

ORDERING INFORMATION

To ensure compatibility between the EPROM/OTP parts and the corresponding ROM families, the following information is provided. The user should take this information into account when programming the memory of the EPROM parts.

Communication of the ROM Codes. To communicate the contents of memories to SGS-THOMSON, the customer has to send:

– one file in INTEL INTELLEC 8/MDS FORMAT (either as an EPROM or in a MS-DOS 5" diskette) for the EEPROM initial content (this file is optional)

– a filled Option List form as described in the OPTION LIST paragraph.

The ROM code must be generated with ST6 assembler. Before programming the EPROM, the buffer of the EPROM programmer must be filled with FFh.

For shipment to SGS-THOMSON the EPROMs should be placed in a conductive IC carrier and packaging carefully.

Customer EEPROM Initial Contents: Format

a. The content should be written into an INTEL INTELLEC format file.

b. In the case of 384 bytes of EEPROM, the starting address is 000h and the end address is 7Fh. The order of the pages (64 bytes each) is an in the specification (ie. b7, b1 b0: 001, 010, 011, 101, 110. 111).

c. Undefined or don't care bytes should have the content FFh.

Listing Generation & Verification. When SGS-THOMSON receives the Codes, they are compared and a computer listing is generated from them. This listing refers extractly to the mask that will be used to produce the microcontroller. Then the listing is returned to the customer that must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed list constitutes a part of the contractual agreement for the creation of the customer mask. SGS-THOMSON sales organization will provide detailed information on contractual points.

ST63E69, T69 MICROCONTROLLER OPTION LIST

Customer: . . . . . .
Address: . . . . . .
Contact: . . . . . .
Phone No: . . . . . .
Reference: . . . . . .

Device [ ] (d) Package [ ] (p) Temperature Range [ ] (t)

For marking one line with 16 characters maximum is possible
Special Marking [ ] (y/n) Line1 "_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _" (N)
(For Plastic Package only)

Notes:
(d) 1= ST63E69, 2 = ST63T69
(p) B= Plastic Dual in Line, D= Ceramic Dual in line with Window
(t) 1= 0 to 70°C
(N) Letters, digits, ' . ', ' - ', ' / ' and spaces only

Marking: the default marking is equivalent to the sales type only (part number).

CHECK LIST:

	YES	NO
EEPROM Code (if Desired)	[ ]	[ ]

Signature .....

Date .....

ORDERING INFORMATION TABLE

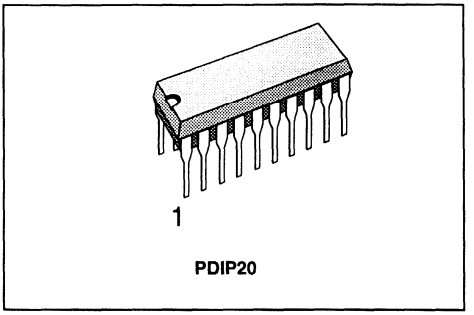
Sales Type	EPROM/EEPROM Size	D/A Converter	Temperature Range	Package
ST63E69D1/XX	8K/384 Bytes	7	0 to + 70 °C	CDIP40W
ST63T69B1/XX	8K/384 Bytes	7	0 to + 70 °C	PDIP40

Note: "XX" Is the ROM code identifier that is allocated by SGS-THOMSON after receipt of all required options and the related ROM file.

ON-SCREEN DISPLAY

ADVANCE DATA

- Display format of 26 characters x 11 rows
- Character Matrix of 12 x 18 pixels
- Character color selectable on a character by character basis. (up to four different colors per screen).
- Character fonts: 128 ROM and 4 RAM based characters.
- Character background selectable on a character by character basis (no background, background 1 or background 2 ; one backround color set per screen).
- Character border (fringe) enable/disable for each row (one border color per screen).
- Programmable vertical and horizontal start position for the display.
- Programmable horizontal offset position for each row.
- Row enable/disable feature.
- Raster Control: The whole screen can be displayed in a color (display off, screen background enabled) or together with the characters (display on, screen background enabled); or transparent (display on/off,screen backround disabled).
- Vertical row spacing: Rows can be spaced or squeezed by up to 17 lines; a squeezed row will have its height reduced by skipping the required number of lines.
- An extra pin MONITOR (enable / disable) is available to indicate the presence of a character pixel or border (fringe).
- Microcontroller interface with a 3 line serial bus
- Oscillator enable/disable.
- Package: 20 pins DIP (300 mils).
- Power Supply: 4.5 to 5.5 volts.



PDIP20

Figure 1. ST6398 Pin Configuration

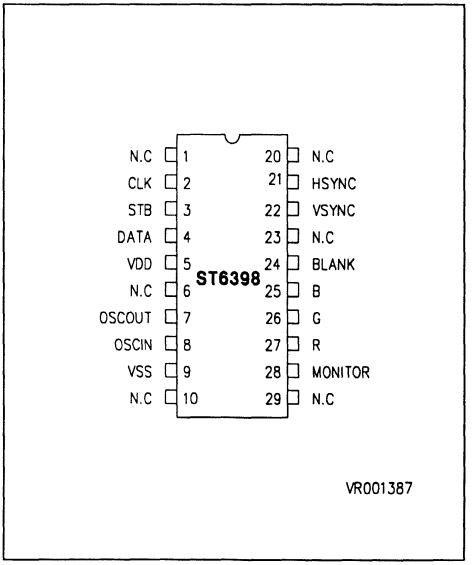
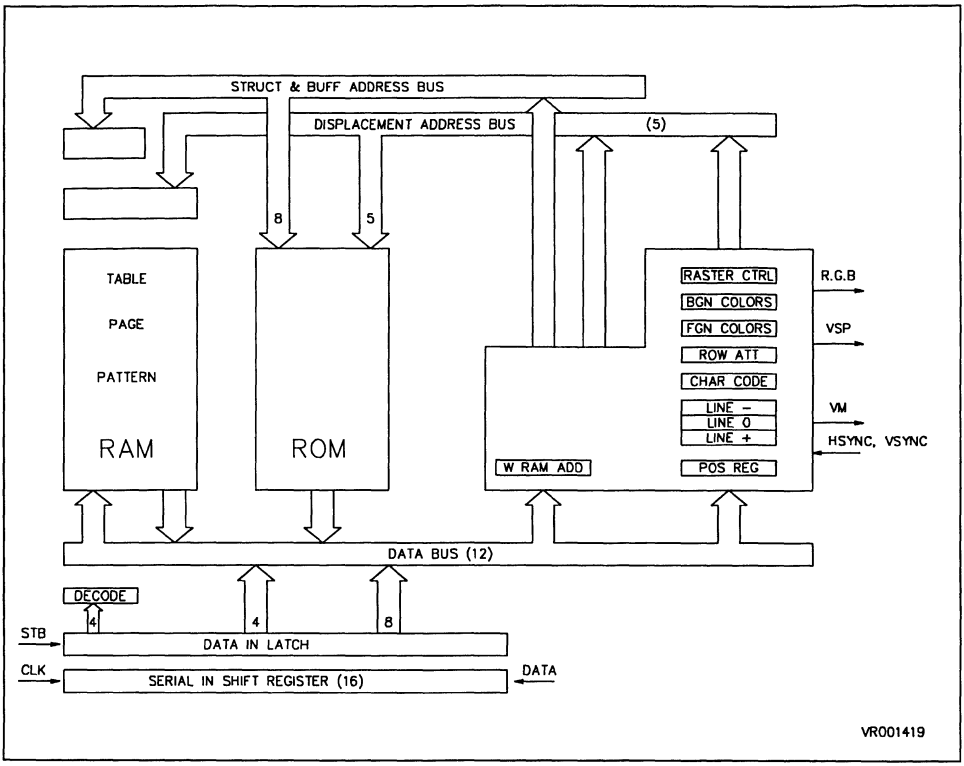


Figure 2. ST6398 Block Diagram



GENERAL DESCRIPTION

The ST6398 is a stand alone on-screen-display peripheral with advanced features.

The device is controlled by an external microcontroller via the serial bus interface. Four characters which can be defined in RAM give great flexibility since characters can be defined and redefined via the serial bus without limit and without changing the character ROM mask. Updating of control registers and characters via the serial bus is not limited to the flyback intervals but can also be performed during the display of characters. Most information is stored in on-chip RAM organized in three main structures: page, attribute and pattern store. The data for the 12-bit words is entered via the serial bus however it is not necessary to redefine all twelve bits via the serial bus when writing similar data to successive addresses.

The device (sales type ST6398B1/B) is delivered with a standard set of ROM based characters; extra characters particular to the application are defined via the RAM characters. If a customized set of ROM based characters is required then contact your SGS-THOMSON sales office.

PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied via these two pins. V_{DD} is power and V_{SS} is the ground connection.

DATA. Serial bus data input; high impedance.

CLK. Serial bus clock input; high impedance.

STB. Serial bus strobe input; high impedance.

OSCIN and OSCOUT. These are the oscillator terminals for the on screen display. A capacitor and coil network should be connected across these pins in order to provide the desired oscillation frequency.

HSYNC. Horizontal synchronization input; high impedance. When the signal goes high the OSD oscillator is started and characters can be displayed. A different polarity can be defined with a customized ROM mask.

VSYNC. Vertical synchronization input, high impedance. When the signal goes high, this indicates the start of the TV display. The OSD oscillator is not affected by this signal. A different polarity can be defined with a customized ROM mask.

R, G, B, BLANK. Color and blanking outputs; push-pull. These outputs are all active high. A different polarity can be defined with a customized ROM mask.

MONITOR. Output pin which indicates the presence of a character pixel or border; push-pull. This output is active LOW. A different polarity can be defined with a customized ROM mask.

NC. The pins are not used in the application and should be left unconnected.

ST6398 ARCHITECTURE OVERVIEW

The ST6398 consists of the following blocks:

- SERIAL MICROPROCESSOR INTERFACE. Most data received through this interface is stored in the on-chip RAM.
- RAM. The RAM holds 12-bit words. It is logically organized in three structures: PAGE, ATTRIBUTE TABLE, PATTERN STORE.
- ROM. This is where the 128 ROM based characters are stored.
- TIMING GENERATOR: This is synchronized by the VSYNC and HSYNC inputs. Its timing is derived from the oscillator (7.5MHz nomimal).
- MEMORY ADDRESS MODULE. Controls the addressing of the RAM and ROM.
- PIXEL LOGIC. Used to determine when to output the color, blanking or monitor information including the backgrounds and border.

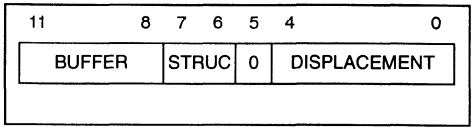
RAM STRUCTURE

RAM Addressing

Random access to any word is performed by a logical 11 bit address, comprised of three fields (see Ram Memory Map):

- A Structure Identifier: 2 bits.
- A Buffer Index: 4 bits.
- A Displacement from the buffer origin: 5 bits.

Figure 3. RAM Addressing



RAM Structure Description

The three structures PAGE, ATTRIBUTE TABLE and PATTERN STORE are divided as follows:

PAGE:

- 11 row buffers, numbered from 0 to 10. Each row buffer holds 26 character words. Inside a buffer, characters words are numbered from 0 to 25; Sequential addressing: Character first, then row buffer.

ATTRIBUTE TABLE:

- 11 row attribute words (0 to 10). Each of the row attribute words (0 to 10) controls the display of the corresponding row buffer.
- The foreground colors (word number 12) which holds the 4 color foreground set (FC0 to FC3).
- The background colors (word number 13) which holds the 2 color character background set (BC0 and BC1), the foreground border color (FBC) and the screen background color (SBC).
- The raster control (word number 14).
- The position register (word number 15) which has been mapped into the RAM for convenience.

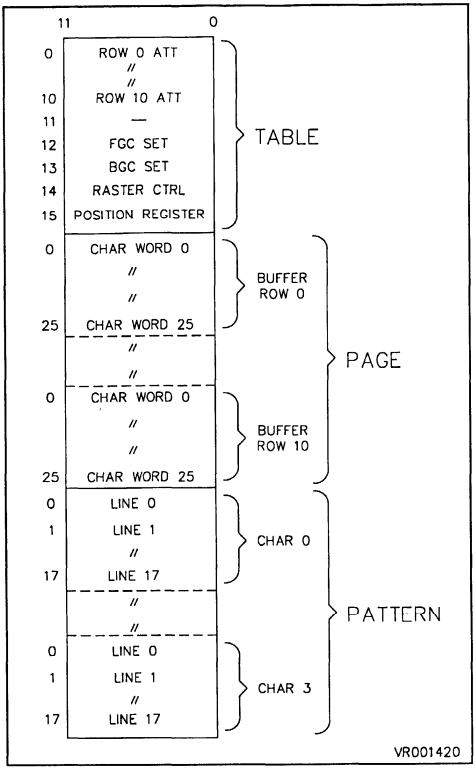
The raster control word, foreground color word, background color word and the current row attribute word are retrieved at the begining of each tv line.

PATTERN STORE:

- Sufficient RAM to describe four character patterns (0 to 3).
  Each character pattern is comprised of 18 pattern lines (0 to 17; one word each). Line 0 is displayed topmost; the msb is displayed leftmost. Sequential addressing: Line first, then character index.

RAM STRUCTURE (Continued)

Figure 4. RAM Memory Map



SERIAL INTERFACE

Microprocessor Interface

The OSD RAM is down loaded from the microcontroller through the three input pins DATA, CLK and STB.

At the rising edge of CLK the DATA input is sampled and shifted in; either 0 bit, 8 bit or 16 bit messages can be received. The MSB is transmitted first.

At the rising edge of STB the number of clock periods counted from last falling edge of STB is sampled; the clock is ignored when STB is high.

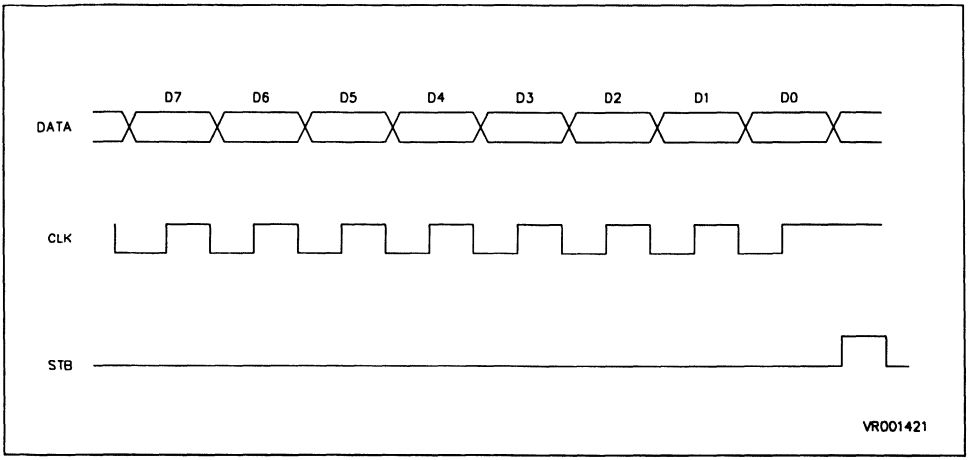
According to this count 0, 8 or 16 bits from the shift register are latched into the 16 bit DATA IN register. Short messages do not modify the most significant bits of DATA IN. The twelve least significant bits of DATA IN can be transfered either to the RAM as DATA or as the WRITE RAM ADDRESS depending on the four most significant bits (4 bit tag).

Transfer to the RAM is done into the location currently pointed to by the WRITE RAM ADDRESS register; this register is automatically post-incremented.

The STB raising edge requests a transfer which is quickly granted providing that the oscillator is running. As the oscillator is blocked during HSYNC, the STB worse case minimum period is related to the maximum HSYNC duration and the oscillator period.

SERIAL INTERFACE (Continued)

Figure 6. Serial Bus Waveforms



Writing To RAM

Writing into RAM consists of two steps:

a. Sending a RAM address.
This selects a starting address in a given structure, together with an auto-incrementation scheme (see auto-incrementation).

b. Sending a string of DATA TO RAM messages. These will be stored sequentially in the selected structure from the starting address.

Long Messages

The 16 bit message consists of:

- A 12 bit word.
- A 4 bit tag (most significant bits).

Figure 5. Summary of 16-Bit Messages

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	BUF				STRUC		0	DISPLACEMENT					RAM ADDRESS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	LONG DATA TO RAM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	DISABLE OSCILLATOR
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	RESET

Auto-incrementation

Sequential addressing is performed by an automatic incrementation of the address after every command that loads data into the RAM; it is therefore not necessary to redefine the address when addressing sequentially.

The auto-incrementation is valid for the TABLE (a single buffer structure which contains the attributes and general control registers) and the PATTERN STORE (a four buffer structure which contains the definition of the RAM based characters).

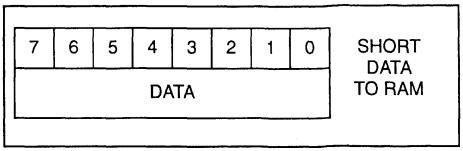
When in PAGE (an 11 buffer structure which defines the characters used on each line), the auto-incrementation can be programmed to wrap-around from the last character of a row either to the begining of the same row or to the begining of next row depending on the value of the two bits that define the structure (as defined the the table below). Refer also to RAM ADDRESSING section.

The TABLE holds a single buffer so auto-incrementation to the next buffer is not applicable. Note that auto-incrementation never allows you to go from one structure to the next one: after the last address of any structure, an illegal address is reached (some are reserved for testing) so AFTER WRITING THE LAST WORD OF ANY STRUCTURE, THE ADDRESS MUST BE REINITIALIZED

Short Data To Ram Messages

In order to minimize the transmission time of character words (or row attributes) differing only by the eight least significant bits, the common four most significant bits may be transmitted only once. This is done by transmitting these MSBs, together with the first word, in a full size 16 bit message (long data to RAM). The subsequent strings may then be compressed by using the short DATA TO RAM message (8 bits).

Figure 7. Short Data to RAM Format



A more radical compression can be achieved when a string of identical words is down loaded into RAM at successive addresses. This is done, as already suggested, by sending the common value once, and then by toggling STB without toggling CLK. This 0 bit message writes the full current content of the DATA IN register into RAM.

Reset And Disable Oscillator Messages

In order to initialize the circuit after power on, it is mandatory to send TWO SUCCESSIVE RESET MESSAGES; this procedure resets the clock count and starts the oscillator.

The oscillator starts regardless of the SYNC input terminals. The RESET messages clear the raster control word (screen background transparent, display disabled). After these two RESET messages the OSD is ready to store subsequent messages. After a DISABLE OSCILLATOR message, the oscillator stops immediately. As long as the oscillator is stopped, the video output terminals remain in their retrace state: RGB, BLANK and MONITOR are all inactive. The RAM contents remain unmodified. When OSD is not in use it is best to disable the oscillator. The oscillator can be restarted by a single RESET message.

Table 1. Auto-Incrementation to Next Buffer Summary

Structure	Buffer Index	Displacement	To Next Buffer by Auto-Incrementation
Page 00	0 to 10	0 to 25	YES
Page 01	0 to 10	0 to 25	NO
Pattern 10	0 to 3	0 to 17	YES
Table 11	0	0 to 15	N/A

RAM DATA FORMAT

Page: Character Word Format

Figure 8. Character Words

11	10	9	8	7	6	5	4	3	2	1	0
FS1	FS0	BE	BS	CHAR INDEX							

CHAR INDEX

0xxxxxxx Addresses one character pattern out of 128 in ROM.

100000xx Addresses one character out of 4 in RAM pattern store.

BE	BS	Character background color selection
1	0	Selects BC0 (in background color word)
1	1	Selects BC1 (in background color word)
0	-	Disables character background; the screen background is therefore displayed if enabled (SBC Screen Background Color in background color word). If SBE = 0 in the raster control word then there is no screen background; if SBE = 1 screen background color is displayed.

FS1	FS0	Character foreground color selection.
0	0	Selects FC0 (in foreground color word).
0	1	Selects FC1 (in foreground color word).
1	0	Selects FC2 (in foreground color word).
1	1	Selects FC3 (in foreground color word).

Note on Blanking:

Whenever a background, border or character is outputed, the BLANK signal will also go high. This is used to externally switch off the normal screen display so that the RGB signals from the OSD can be displayed.

Table

ROW ATTRIBUTE WORDS (numbered 0 to 10).

Figure 9. Row Attribute Word

11	10	9	8	7	6	5	4	3	2	1	0
HPOS OFFSET				RE	FBE	VPOS OFFSET					

HPOS OFFSET: Adds a horizontal offset (0 to 15 Tosc) to the basic horizontal position for any line belonging to the row (bit 8 is lsb).

VPOS OFFSET:

0xxxxx Adds n (0 to 17) tv lines at the top of the plain 18 line row. These lines are displayed in the character background color (as defined by the current BE, BS) during character period. During the horizontal front and back porch period these lines are displayed in the screen background if enabled or transparent if disabled (bit 0 is lsb).

1xxxxx Jumps directly to line n (0 to 17) in the display of that particular row. This means that the row height is reduced by n lines (bit 0 is lsb).

FBE:

0 Border (fringe) effect disabled.

1 Border (fringe) effect enabled; any background pixel in the vicinity of a foreground pixel is diplayed in foreground border color (fetched from the background color word).

RE:

0 The row is disabled; nothing is displayed except the screen background color if enable; VPOS OFFSET is still interpreted.

1 Row enable

RAM DATA FORMAT (Continued)

FOREGROUND COLOR WORD (number 12)

Figure 10. Foreground Color Word

11	10	9	8	7	6	5	4	3	2	1	0
FC3			FC2			FC1			FC0		

FC0 to FC3: The foreground color word holds the character foreground color set.

Each of the above defines an RGB triplet (B is the LSB):

RGB	COLOR
000	Black
001	Blue
010	Green
011	Cyan
100	Red
101	Magenta
110	Yellow
111	White

BACKGROUND COLOR WORD (number 13)

Figure 11. Background Color Word

11	10	9	8	7	6	5	4	3	2	1	0
SBC			FBC			BC1			BC0		

BC0, BC1: Defines the background color set definitions.

FBC: Defines the character border (fringe) color.

SBC: Defines the screen background color.

Each of the above defines an RGB triplet (B is the LSB):

RGB	COLOR
000	Black
001	Blue
010	Green
011	Cyan
100	Red
101	Magenta
110	Yellow
111	White

RASTER CONTROL WORD (number 14).

Figure 12. Raster Control Word

11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	DON	FME	SBE

SBE: Screen Background Enable.

0 Screen background is off; the normal video image is enabled.

1 The screen background is enabled; the video image is disabled by the BLANK signal (raster blanking) and the screen is forced to the color defined by SBC (screen background color).

FME: Foreground Monitoring Enable.

0 Disabled. MONITOR output terminal stays active.

1 Enabled. MONITOR output terminal is activated only at the time when a character pixel or border (fringe) is being displayed; it is inactive in all other cases (eg. background display).

DON:

0 The display is OFF; all the characters are disabled; the full screen displayed as either "transparent" or "blanked" depending on the state of SBE.

1 The OSD character display is ON.

POSITION REGISTER (number 15).

Figure 13. Raster Control Word

11	10	9	8	7	6	5	4	3	2	1	0
VPOS						HPOS					

VPOS: Vertical start position; counts 1 to 63 lines from the top of the screen.

HPOS: Horizontal start position; counts 5 to 63 oscillator periods from the left of the screen.

NOTE: Correct operation is not guaranteed if these parameters are programmed under the minimum values given above.

RAM DATA FORMAT (Continued)

PATTERN WORDS

Figure 14. Pattern Words

11	10	9	8	7	6	5	4	3	2	1	0
P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11

Word 0 in the pattern buffer is displayed topmost on the screen.

Most significant bit is displayed on the left of the screen.

The foreground pixels are set to 1.

The background pixels (ie. no pixel) are 0.

This set of rules also applies to the ROM pattern store.

VIDEO TIMING

Introduction

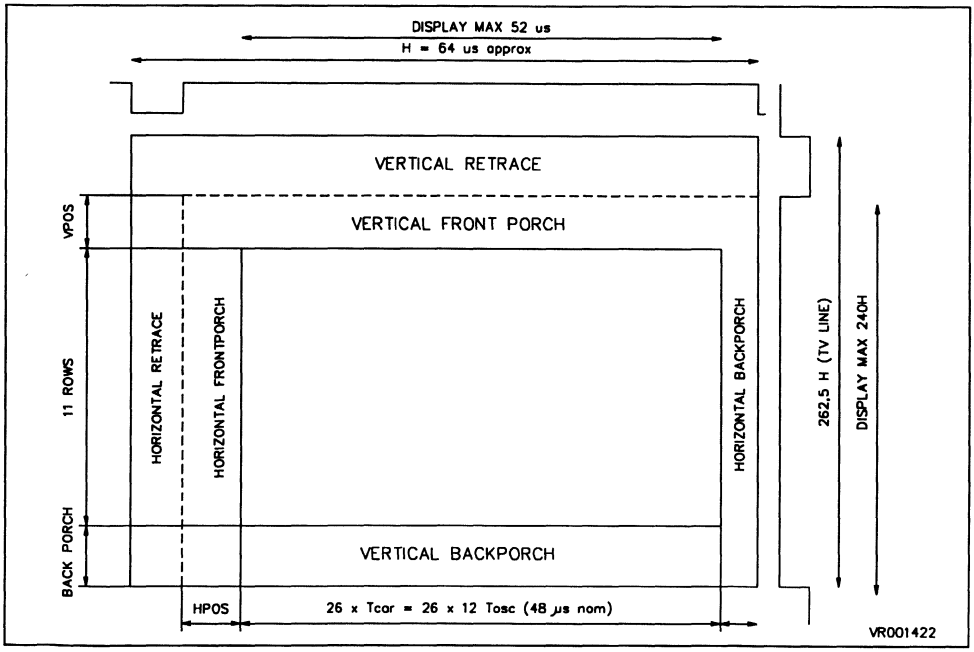
The OSD cycles through three main states: RETRACE, PORCH and CHARACTER.

The vertical and horizontal RETRACE are determined by the VSYNC and HSYNC signals.

The vertical and horizontal FRONT PORCH are determined by the POSITION REGISTER. For the vertical front porch, VPOS horizontal lines are counted down from the VSYNC trailing edge. For the vertical front porch HPOS + HPOS OFFSET clock periods are counted down from the HSYNC trailing edge.

The CHARACTER state is when the 11 rows of 26 CHARACTERS are interpreted. The horizontal and vertical BACK PORCH occur after the last character or row respectively.

Figure 15. Typical OSD Display



VIDEO TIMING (Continued)

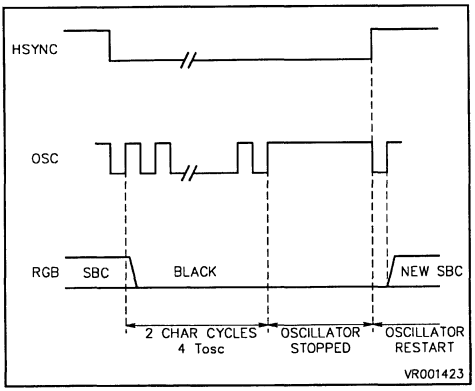
Horizontal Timing

The leading edge of HSYNC is resynchronized at TOSC and the 24 subsequent oscillator periods (2 character cycles) are used to fetch the current row attribute word, the foreground color word, the background color word and the raster control word. The oscillator is then is stopped.

HSYNC lasts nominally for 10us depending on the application. When enabled, the oscillator is restarted at the trailing edge of HSYNC. HPOS + HPOS ROW OFFSET oscillator periods are counted during this HORIZONTAL FRONT PORCH period. After this the 26 CHARACTER periods (12 Tosc each) are counted. The HORIZONTAL BACKPORCH period occurs after the last character while waiting for the next HSYNC.

If HSYNC arrives before the end of the 26th character, then there will be no backporch; HSYNC will restart full line process.

Figure 16. Horizontal Timing During Retrace



Vertical Timing

VSYNC marks the vertical retrace. It lasts nominally for 0.9 ms (14 TV lines) but will depend on the application. The VSYNC signal has no influence on the oscillator and does not start or stop it.

VPOS lines (leading HSYNC edges) are counted down during VERTICAL FRONT PORCH PERIOD. VPOS is a 6 bit binary parameter.

During the display of the eleven rows, the number of lines per row depends on the VPOS OFFSET row attribute. The VERTICAL BACKPORCH period occurs after the last row while waiting for the next VSYNC.

Ram Time Sharing

The RAM is cycled in 2 tosc (300 ns approximately) so that six access time slots can be provided along any character display period (12 tosc):

Four slots are reserved for character interpretation: one slot for fetching the character word and three slots for fetching three successive character lines in the pattern store (this is required for deriving the foreground border when the character pattern is stored in RAM).

The two remaining slots are available to store a serially received word.

VIDEO OUTPUT

Retrace And Porch

The video terminal outputs RGB, BLANK and MONITOR are driven according to:

- The video timing state: RETRACE, PORCH or CHARACTER;
- The programmed context retrieved from RAM.

DURING RETRACE

- RGB are inactive.
- BLANK and MONITOR are inactive. These values prevail also during the full field when the oscillator has been disabled.

DURING PORCH

- RGB are inactive unless the screen background is enabled (SBE=1) in which case RGB will set to the screen background color (SBC).
- BLANK is inactive unless the screen background is enabled (SBE=1) in which case it will be active.
- MONITOR is inactive. These values prevail also during the character period when the display is disabled (DON =0) or when the current row is disabled (RE=0 in the current row attribute word).

Character State

CHARACTER DISPLAY

During a character display, the video terminal outputs are driven according to the pixel type, the current row attribute word, the foreground and background color words , the raster control word and the oscillator status. In this section the following is assumed: oscillator enabled, display enabled, row enabled.

The pattern indexed by the character word is mapped onto the screen and determines the pixel type:

- Foreground. Active when the bit in the pattern store is set to 1.
- Background: When the bit in the pattern store is reset to 0 and when not in the vicinity of a foreground pixel, then the background will be displayed. When VPOS adds n tv lines to the row, these extra lines will be implicitly displayed as background pixels.
- Border (or fringe): The border occurs when the bit in the pattern store is reset to 0, when in the vicinity of at least one foreground pixel and when FBE=1 in the row attribute word. If FBE=0 then there is no border. The border will outline the shape of the character.

VICINITY

Each pixel has four neighbours (two vertical, two horizontal). The vicinity is horizontally continuous from character to character so that the first column of a charactacter neighbours the last column of the previous character. Vicinity is not vertically cointinuous between different lines. Refer to the following diagram for examples on vicinity in order to see how the border is formed.

Figure 17. Examples of Borders

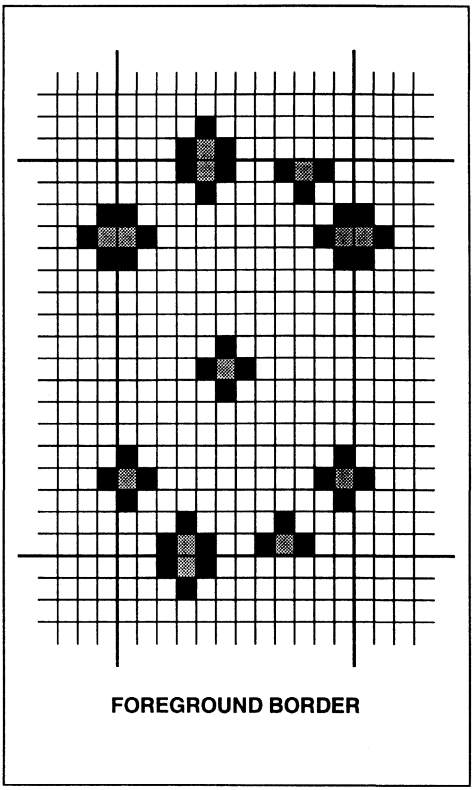


Table 2. The Different Character States

Type of Pixel Condition	Outputs		
	RGB	Blank	Monitor *
Background BE = 1 BE = 0, SBE = 1 BE = 1, SBE = 0	 Background color Screen background Inactive (Black)	 Active Active Inactive	 Inactive Inactive Inactive
Foreground	Character color	Active	Active
Border FBE = 1	Border color (FBC)	Active	Active

Note : * Assumes VME = 1; when VME = 0, MONITOR remains active.

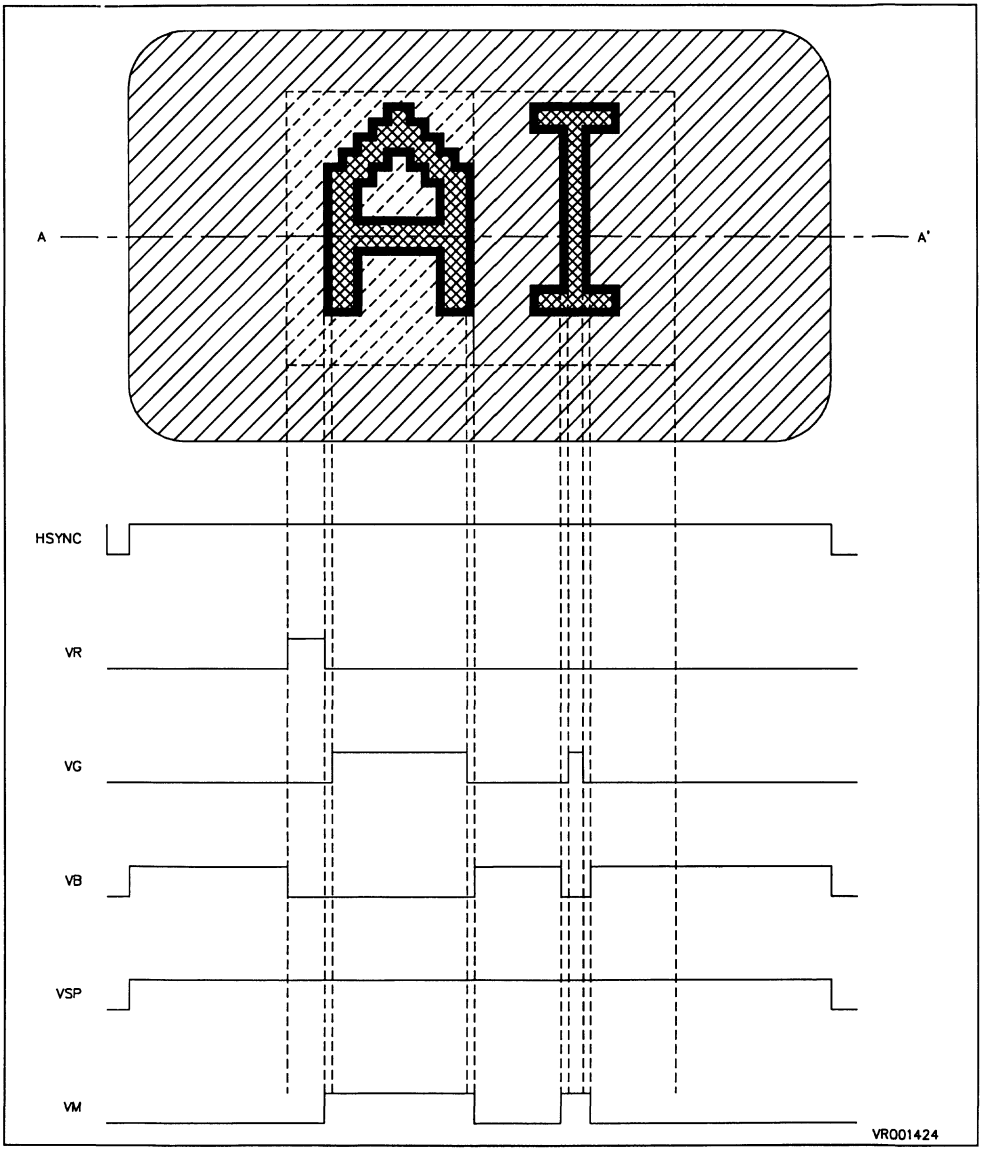
OSD WAVEFORM EXAMPLES

Note: for the following examples, the following polarities have been assumed:

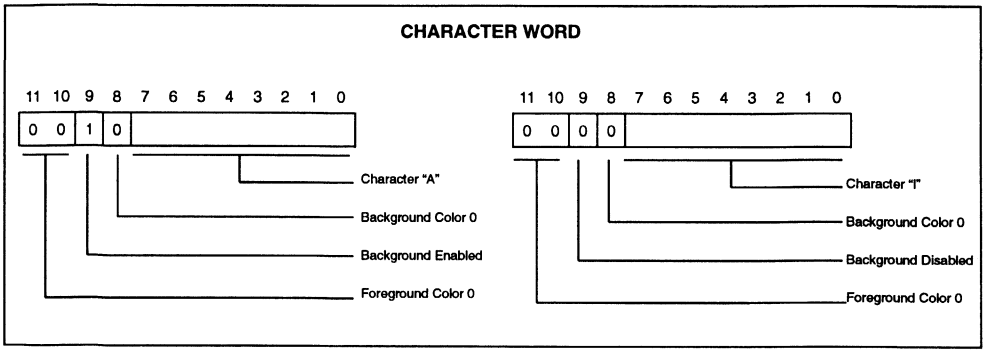
HSYNC,VSYNC . . . . . negative

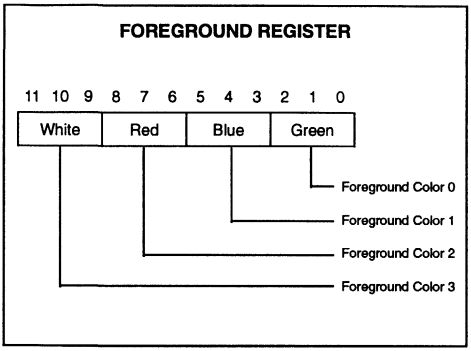
R,G,B,BLANK,MONITOR . . . . . positive

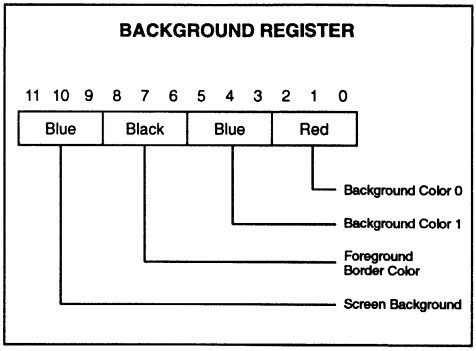
Figure 18.

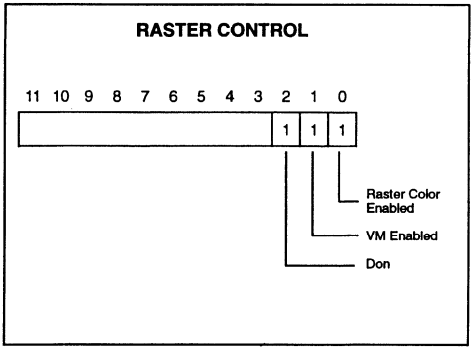


OSD WAVEFORMS EXAMPLES (Continued)



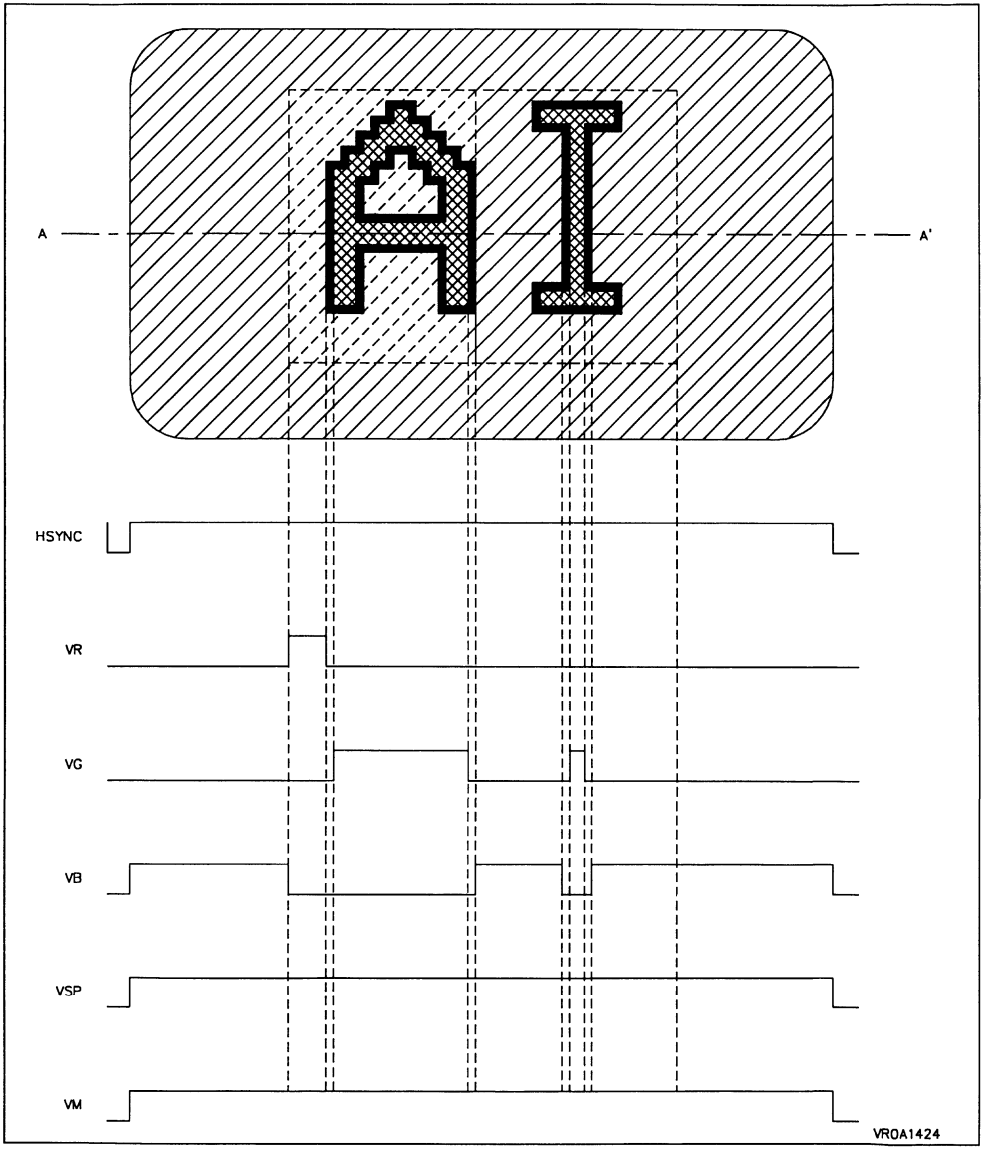




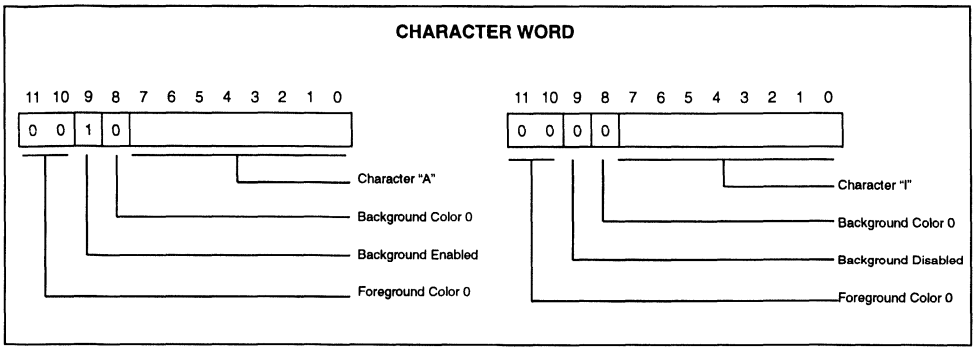


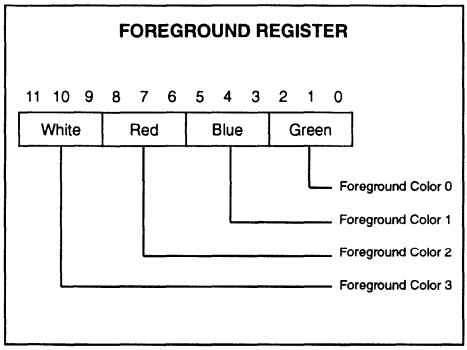
OSD WAVEFORM EXAMPLES (Continued)

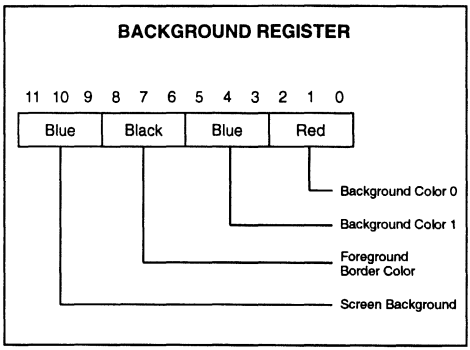
Figure 19.

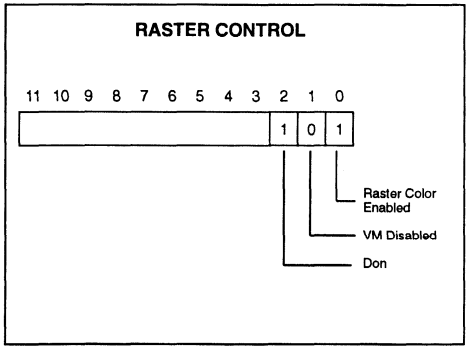


OSD WAVEFORMS EXAMPLES (Continued)



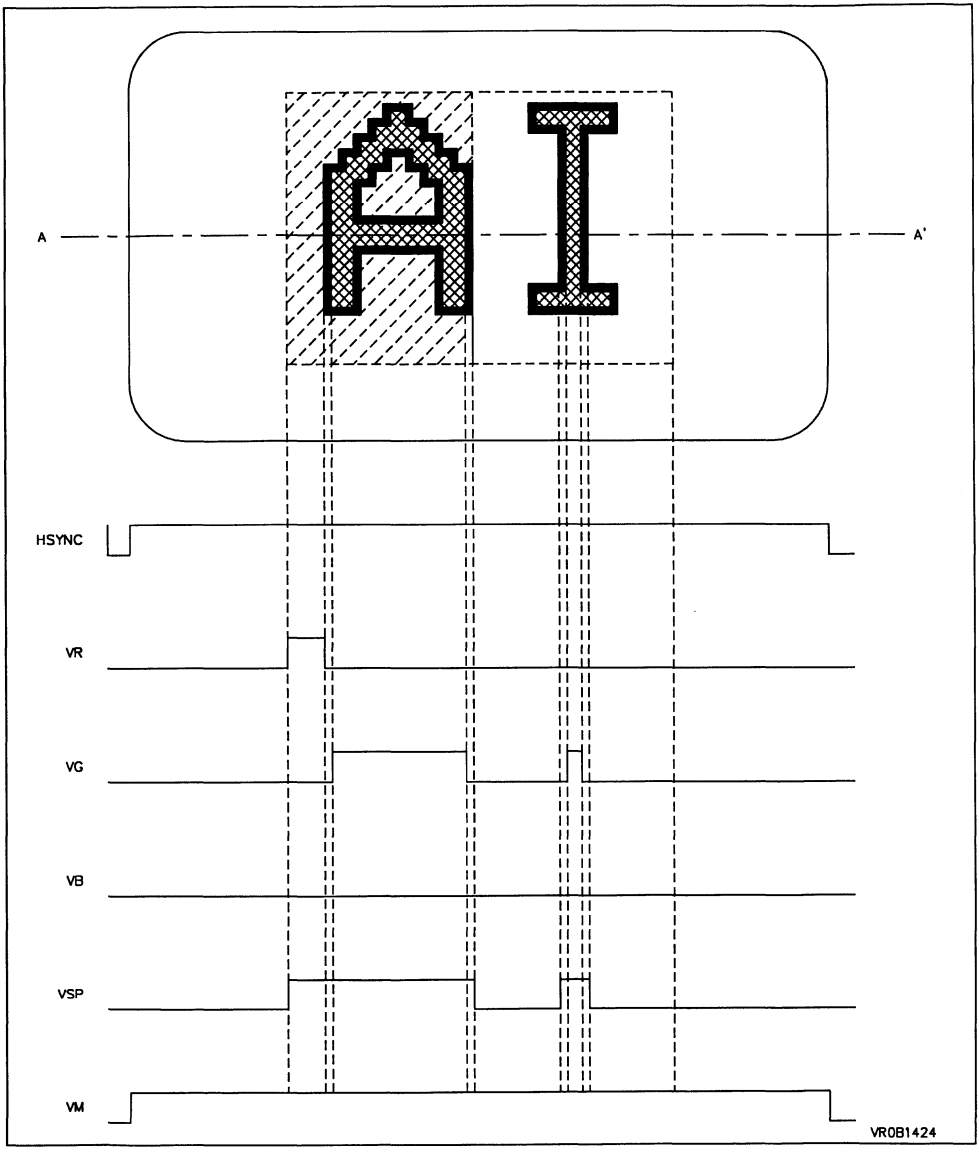




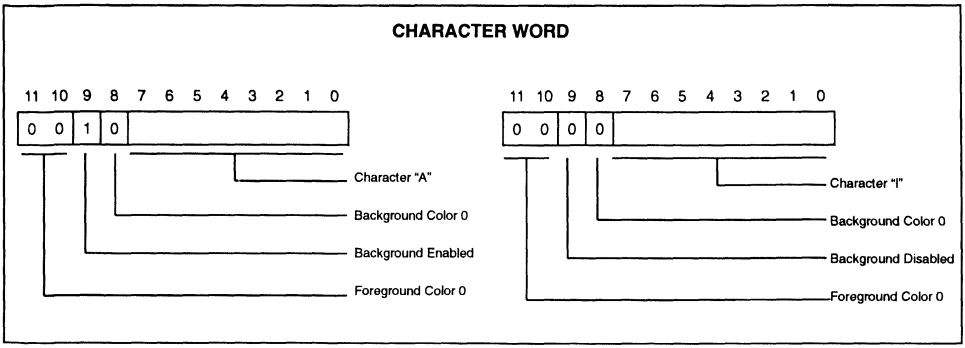


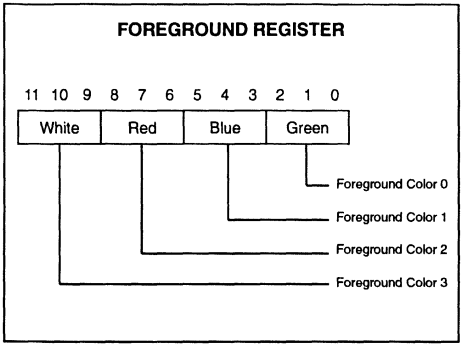
OSD WAVEFORM EXAMPLES (Continued)

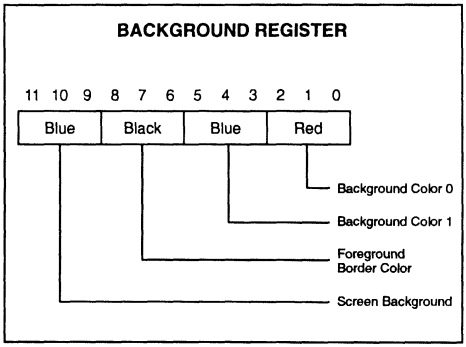
Figure 20.

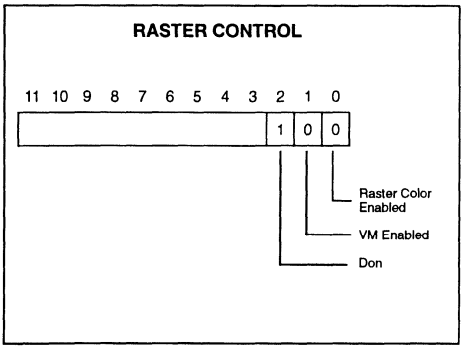


OSD WAVEFORMS EXAMPLES (Continued)









STANDARD CHARACTERS

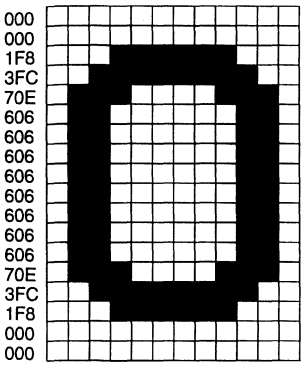
The default device (sales type ST6398B1/B) is delivered with a standard set of ROM based characters and masked pin polarities; extra characters particular to the application are defined via the RAM characters.

The pin polarities have been fixed as follows:

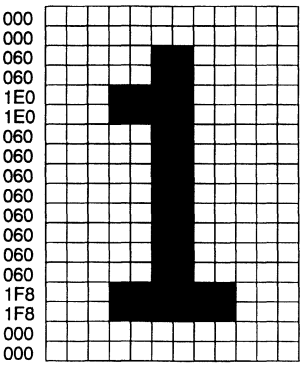
HSYNC,VSYNC	negative (input low during retrace)
R,G,B,BLANK	positive
MONITOR	negative

Forty standard alphanumeric characters has been defined in character locations 00 to 27h; the remaining characters up to address 7Fh are not defined. Should these characters (together with the user definable and variable RAM characters) not be sufficient for the customer's application, then a customized set can be defined (see the section on customized characters). The standard set of ROM based characters is fixed as shown below.

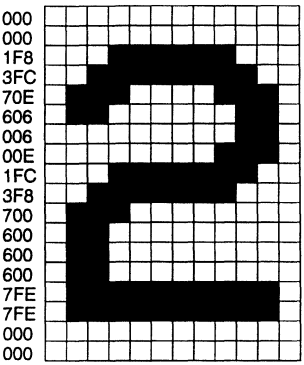
CHARACTER DEFINITION



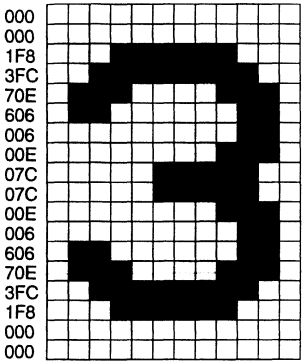
00



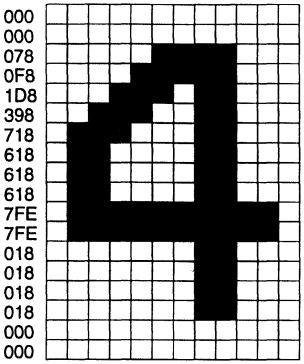
01



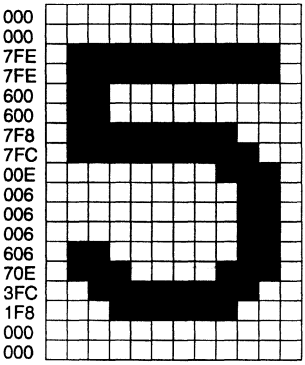
02



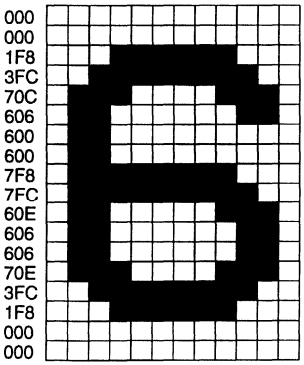
03



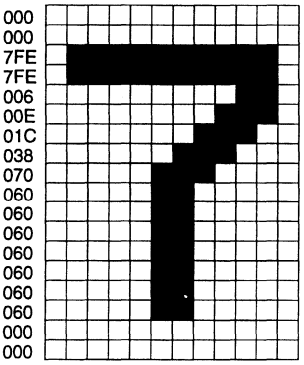
04



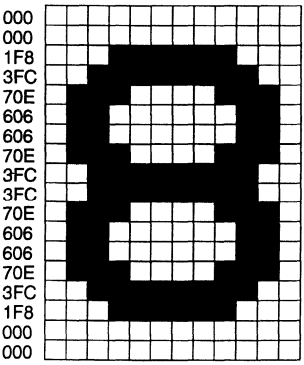
05



06

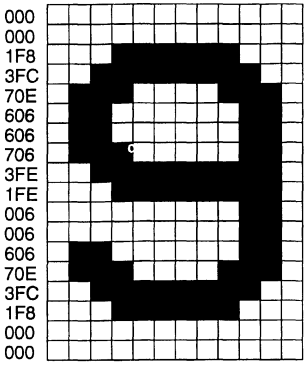


07

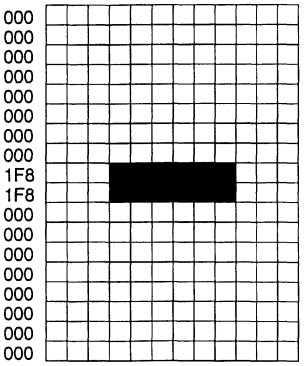


08

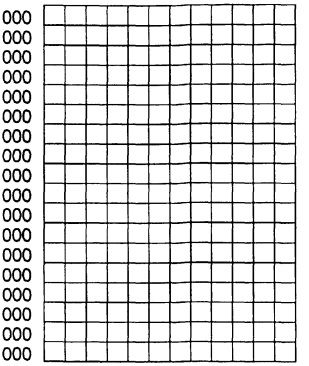
CHARACTER DEFINITION (Continued)



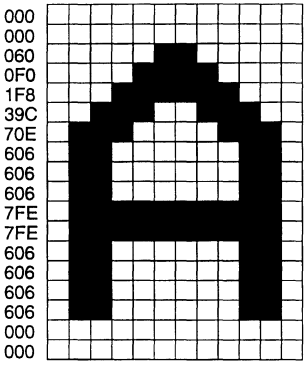
09



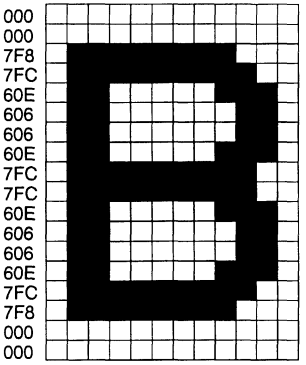
0A



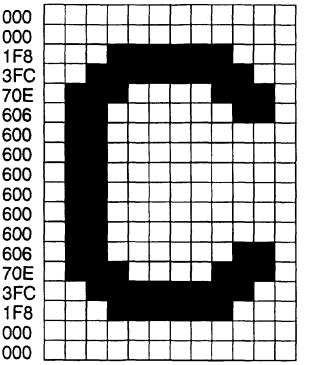
0B



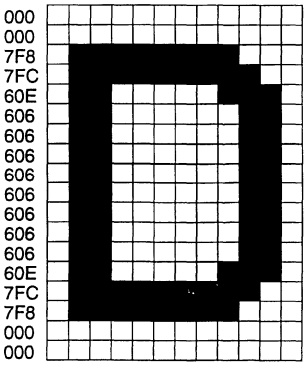
0C



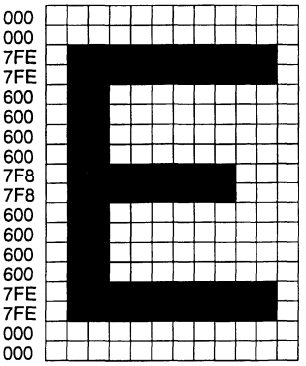
0D



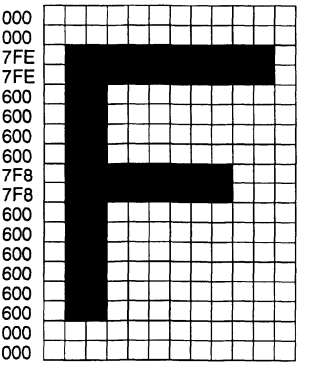
0E



0F

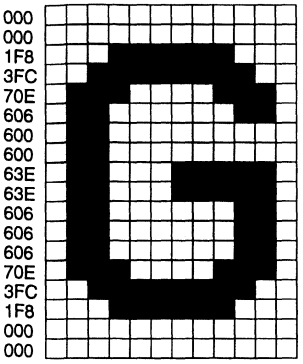


10

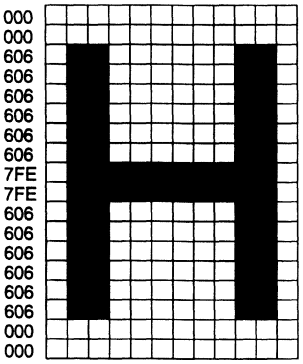


11

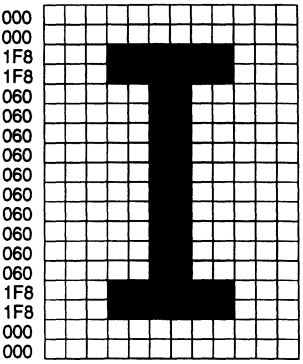
CHARACTER DEFINITION (Continued)



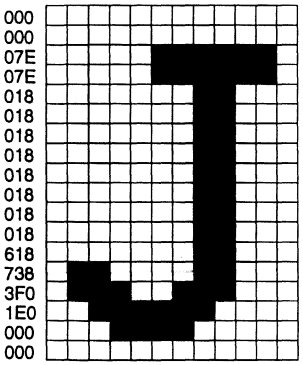
12



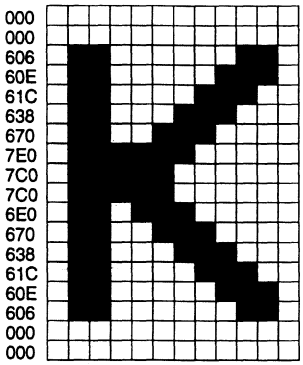
13



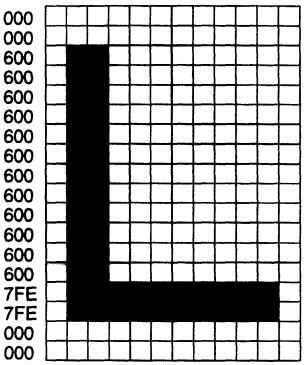
14



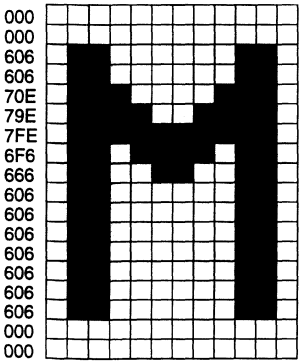
15



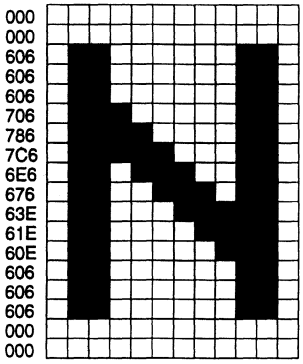
16



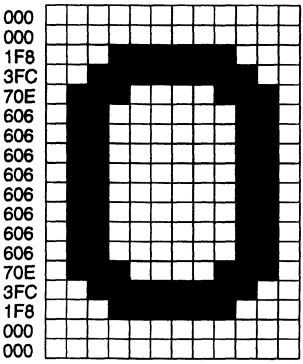
17



18

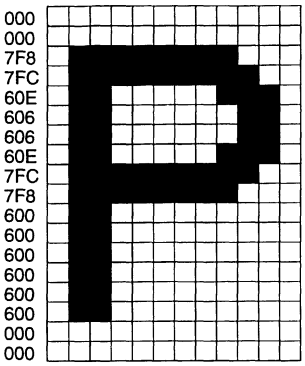


19

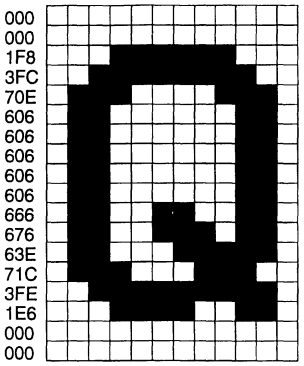


1A

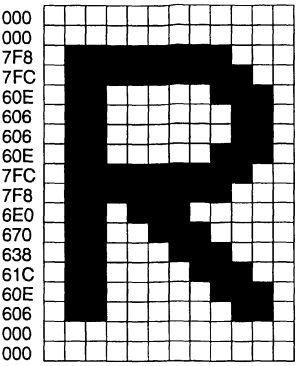
CHARACTER DEFINITION (Continued)



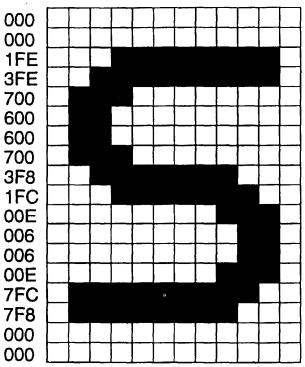
1B



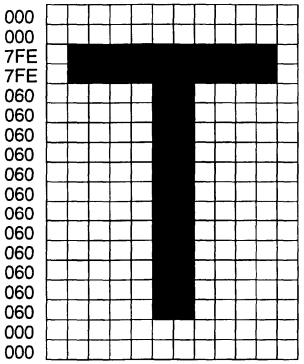
1C



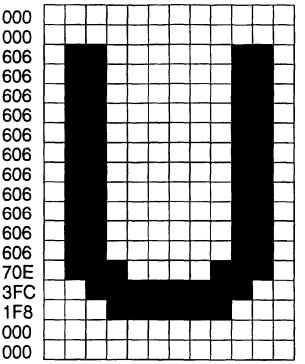
1D



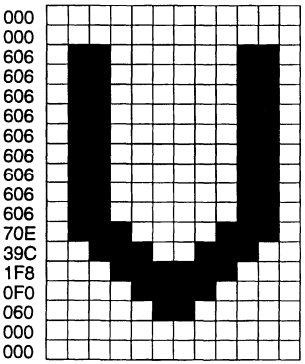
1E



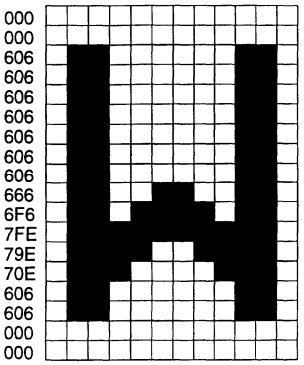
1F



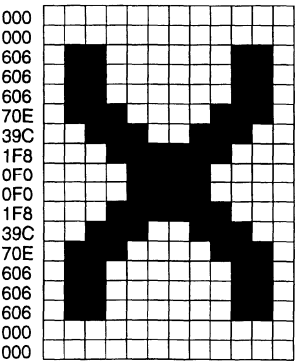
20



21

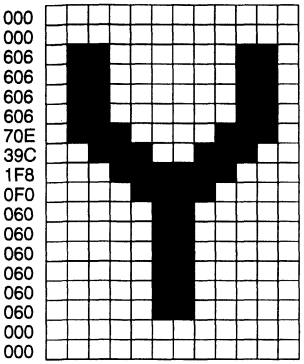


22

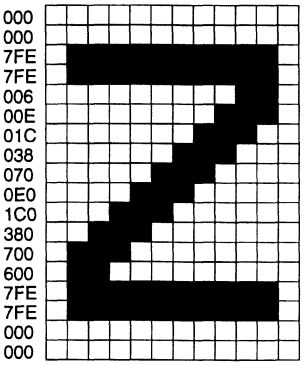


23

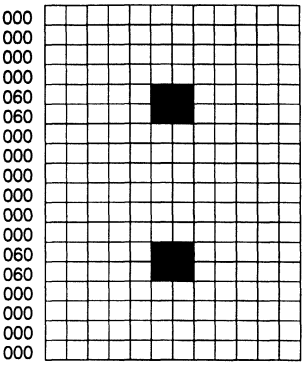
CHARACTER DEFINITION (Continued)



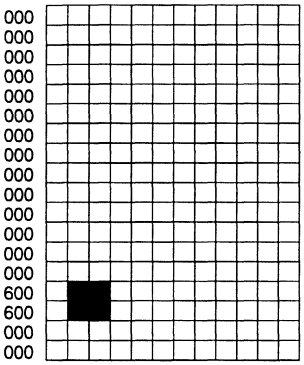
24



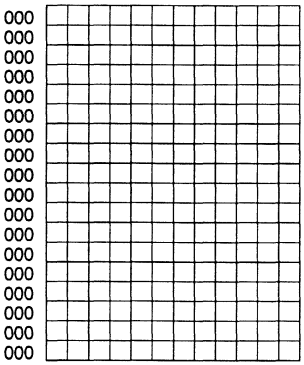
25

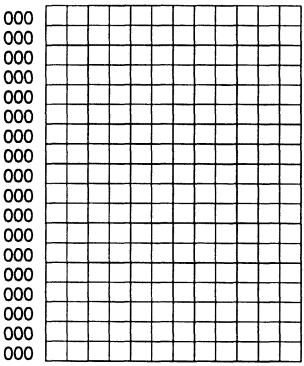


26



27





ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages. For proper operation it is recommended that V_I and V_O must be high than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Symbol	Parameter	Value		Unit
		Min.	Max.	
V_{DD}	Supply Voltage	-0.3	7	V
V_{IN}	Input Voltage	V_{SS} - 0.3	V_{DD} + 0.3	V
V_{OUT}	Output Voltage	V_{SS} - 0.3	V_{DD} + 0.3	V
T_J	Junction Temperature		150	°C
T_{STG}	Storage Temperature	-60	150	°C
I_O	Current per pin	-5	+5	mA

Note: Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Value			Unit
		Min.	Typ.	Max.	
V_{DD} - V_{SS}	Supply Voltage	4.5	5.0	5.5	V
f_{OSC}	Oscillation Frequency	4.0	7.5	8.0	MHz
f_{CLK}	Clock Frequency	0		2	MHz
T_{OPT}	Operating Temperature	0		70	°C

THERMAL CHARACTERISTICS

Symbol	Parameter	Test Condition	Value			Unit
			Min.	Typ.	Max.	
Rth_{JA}	Thermal Resistance	DIP20			60	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, Tj, in Celsius can be obtained from:

$T_J = T_A + P_D \times Rth_{JA}$

Where: T_A = Ambient Temperature,
Rth_{JA} = Package thermal resistance (junction-to ambient),
P_D = Pint + Pport,
Pint = $I_{DD} \times V_{DD}$ (chip internal power),
Pport = Port power dissipation (determined by the user).

DC ELECTRICAL CHARACTERISTICS

(T_A = 0 to 70°C, V_{DD} = 4.5V to 5.5V unless otherwise specified)

Symbol	Parameter	Condition	Value		Unit
			Min.	Max.	
I_{DD}	Current Consumption			15	mA
V_{IH}	Input High Level		0.75xV_{DD}		V
V_{IL}	Input Low Level			0.15xV_{DD}	V
V_{OH}	Output High Level	I_{OH} = -1.0mA	V_{DD} - 0.5		V
V_{OL}	Output Low Level	I_{OL} = 1.0mA		0.5	V
I_{IL}	Input Low Leakage Current	V_{IL} = 0V	-10		µA
I_{IH}	Input High Leakage Current	V_{IH} = V_{DD}		10	µA

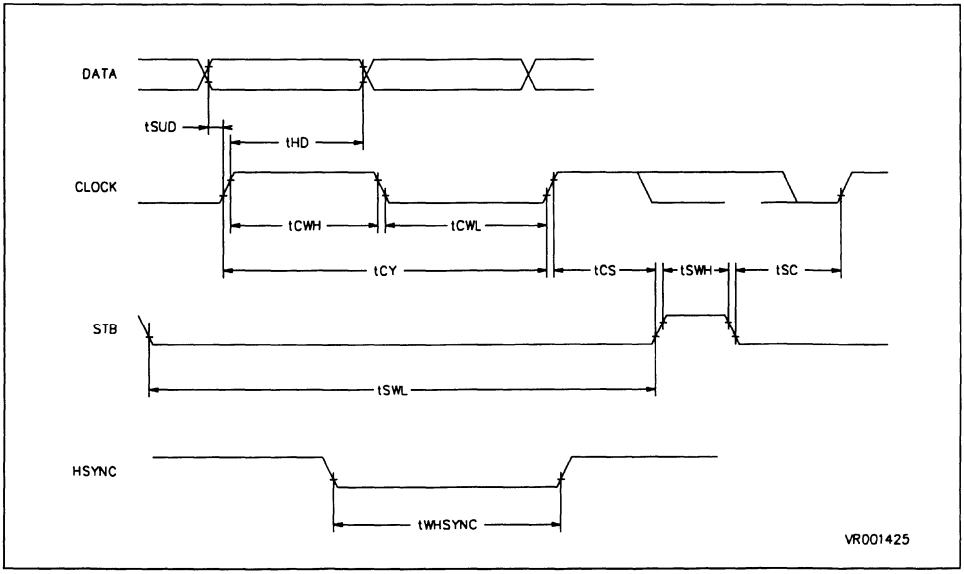
AC ELECTRICAL CHARACTERISTICS

(T_A = 0 to 70°C, V_{DD} = 4.5V to 5.5V unless otherwise specified)

Symbol	Parameter	Value		Unit
		Min.	Max.	
t_{CY}	Clock Cycle	500		ns
t_{CWH}	Clock High	200		ns
t_{CWL}	Clock Low	200		ns
t_{SUD}	Data Set Up	100		ns
t_{HD}	Data Hold	100		ns
t_{CS}	Clock to Strobe	200		ns
t_{SWH}	Strobe High	t_{WHSYNC}		tosc
t_{SWL}	Strobe Low (1)	t_{WHSYNC}		tosc
t_{SC}	Strobe to Clock	200		ns
t_{WHSYNC}	HSYNC Width	50		tosc

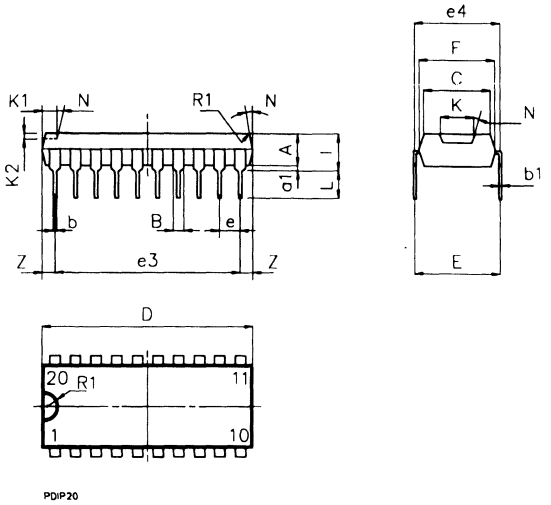
Note 1. For a 0 bit message.

Figure 21. AC Electrical Characteristics



PACKAGE MECHANICAL DATA

Figure 22. 20 Lead Plastic Dual In Line Package (B)



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	–	–	–	–	–	–
a1	0.254			0.010		
B	1.39		1.65	0.054		0.064
b		0.45			0.017	
b1		0.25			0.009	
C	–	–	–	–	–	–
D			25.4			1.000
E		8.5			0.334	
e		2.54			0.100	
e3		22.86			0.900	
e4	–	–	–	–	–	–
F			7.1			0.279
I			3.93			0.154
L		3.3			0.129	
Z			1.34			0.052

CUSTOMED CHARACTERS AND ORDERING INFORMATION

Character Definition

If necessary Customized characters can be developed using the ST6398 CHARACTER DEVELOPMENT BOARD (emulator board). The board is inserted into a PC and is used to develop and simulate the OSD characters and functions. The graphics system is built around the alphanumeric and graphics contoller TS68483 which contains a separate dedicated display memory.

For ST6398 CHARACTER DEVELOPMENT BOARD (emulator board) ordering information contact your SGS-THOMSON sales representative.

Code Procedure

Upon completion of character development, the file created by the SAVE command should be given to SGS-THOMSON (on a MS-DOS 5" diskette) together with the option list.

When SGS-THOMSON receives the file, a character listing is generated and returned to the customer together with a copy of the option list for approval. The signed list constitutes a part of the contractual agreement for the creation of the customer mask. A mask charge together with a minimum quantity of parts will be charged to the customer. The SGS-THOMSON sales organization will provide detailed information on contractual points.

ORDERING INFORMATION

Type	Description	Frequency	Range	Package
ST6398B1	On Screen Display	2MHz	0 to +70°C	DIP20

ST6398 MICROCONTROLLER OPTION LIST

Customer : . . . . . .

Address: . . . . . .

Contact: . . . . . .

Phone No: . . . . . .

Reference: . . . . . .

Package [ ] (p) Temperature Range [ ] (t)

For marking one line with 10 characters maximum is possible

Special Marking [ ] (y/n) Line1 " _ _ _ _ _ _ _ _ _ _ " (N)

Notes:

(p) B= Dual in Line Plastic

(t) 1= 0 to 70°C

(N) Letters, digits, ' . ', ' - ', ' / ' and spaces only

Marking: the default marking is equivalent to the sales type only (part number).

OSD POLARITY OPTIONS (Put a cross on selected item) :

		POSITIVE	NEGATIVE
VSYNC		[ ]	[ ]
HSYNC		[ ]	[ ]
R	(VR)	[ ]	[ ]
G	(VG)	[ ]	[ ]
B	(VB)	[ ]	[ ]
BLANK	(VSP)	[ ]	[ ]
MONITOR	(VM)	[ ]	[ ]

Signature .....

Date .....

Note: A negative SYNC means the input is low during retrace.

Pages

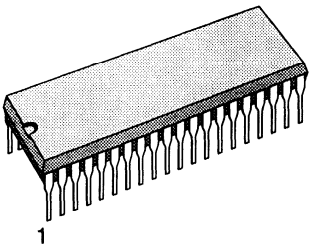
SGS-THOMSON MICROELECTRONICS

ST63140, ST63142
ST63126, ST63156

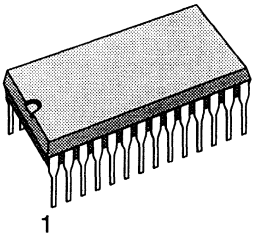
8-BIT HCMOS MCUs FOR TV FREQUENCY & VOLTAGE SYNTHESIS WITH OSD

PRELIMINARY DATA

- 4.5 to 6V operating Range
- 8MHz Maximum Clock Frequency
- User Program ROM: 7948 bytes
- Reserved Test ROM: 244 bytes
- Data ROM: user selectable size
- Data RAM: 256 bytes
- Data EEPROM: 128 bytes
- 40-Pin Dual in Line Plastic Package for the ST63126, 156
- 28-Pin Dual in Line Plastic Package for the ST63140, 142
- Up to 18 software programmable general purpose Inputs/Outputs, including 8 direct LED driving Outputs
- 3 Inputs for keyboard scan (KBY0-2)
- Up to 4 high voltage outputs (BSW0-3)
- Two Timers each including an 8-bit counter with a 7-bit programmable prescaler
- Digital Watchdog Function
- Serial Peripheral Interface (SPI) supporting S-BUS/ I^2C BUS and standard serial protocols
- Up to Four 6-bit PWM D/A Converters
- 62.5kHz Output pin
- 14 bit counter for voltage synthesis tuning (ST63156, ST63140)
- AFC A/D converter with 0.5V resolution
- Four interrupt vectors (IRIN/NMI, Timer 1 & 2, VSYNC.)
- On-chip clock oscillator
- 5 Lines by 15 Characters On-Screen Display Generator with 128 Characters (2 banks)
- All ROM types are supported by pin-to-pin EPROM and OTP versions.
- The development tool of the ST631xx microcontrollers consists of the ST63TVS-EMU emulation and development system to be connected via a standard RS232 serial line to an MS-DOS Personal Computer.



PDIP40



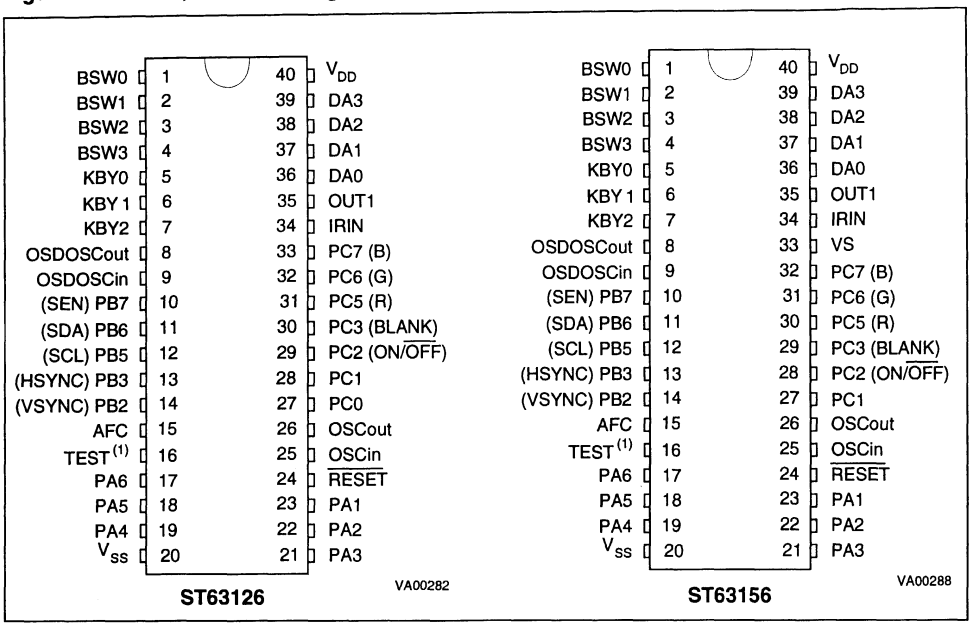
PDIP28

(Ordering Information at the end of the datasheet)

DEVICE SUMMARY

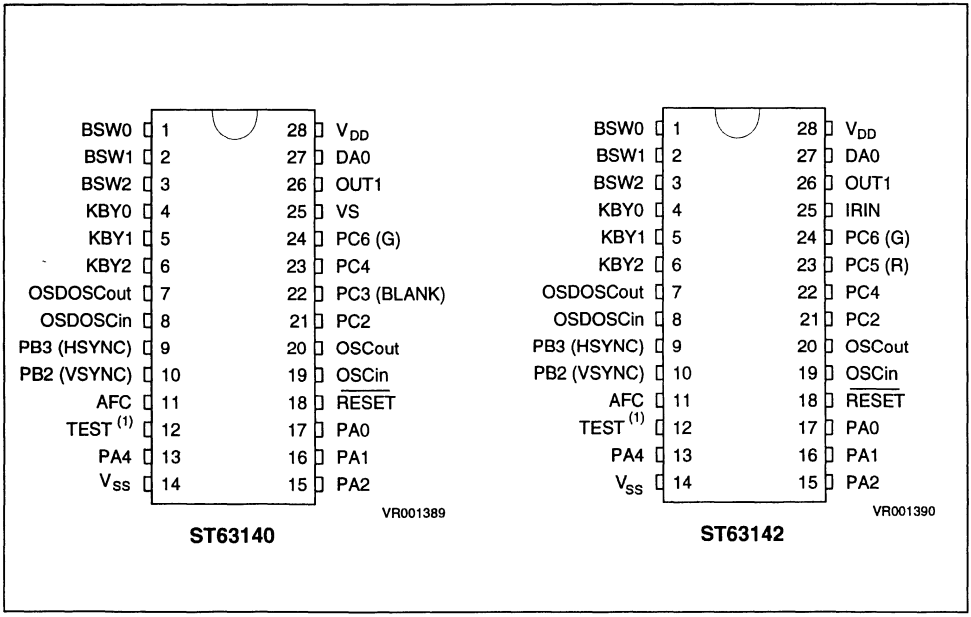
DEVICE	ROM (Bytes)	TUN.	I/O Pins	Package
ST63126	8K	FS	12	PDIP40
ST63156	8K	VS	11	PDIP40
ST63140	8K	VS	6	PDIP28
ST63142	8K	FS	6	PDIP28

Figure 1. ST63126, 156 Pin Configuration



Note 1. This pin is also the VPP input for EPROM based devices

Figure 2. ST63140, 142 Pin Configuration



Note 1. This pin is also the VPP input for EPROM based devices

GENERAL DESCRIPTION

The ST63140, 142, 126, 156 microcontrollers are members of the 8-bit HCMOS ST631xx family, a series of devices specially oriented to TV applications. Different ROM size and peripheral configurations are available to give the maximum application and cost flexibility. All ST631xx members are based on a building block approach: a common core is surrounded by a combination of on-chip peripherals (macrocells) available from a standard library. These peripherals are designed with the same Core technology providing full compatibility and short design time. Many of these macrocells are specially dedicated to TV applications. The macrocells of the ST631xx family are: two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), a digital hardware activated watchdog function (DHWD), a 14-bit voltage synthesis tuning peripheral, a Serial Peripheral Interface (SPI), up to four 6-bit PWM D/A converters, an AFC A/D converter with 0.5V resolution, an on-screen display (OSD) with 15 characters per line and 128 characters (in two banks each of 64 characters). In addition the following memory resources are available: program ROM (7K), data RAM (256 bytes), EEPROM (128 bytes).

Refer to pin configuration figures and to ST631xx device summary (Table 1) for the definition of ST631xx family members and a summary of differences among the different types.

Figure 3. ST631xx Block Diagram

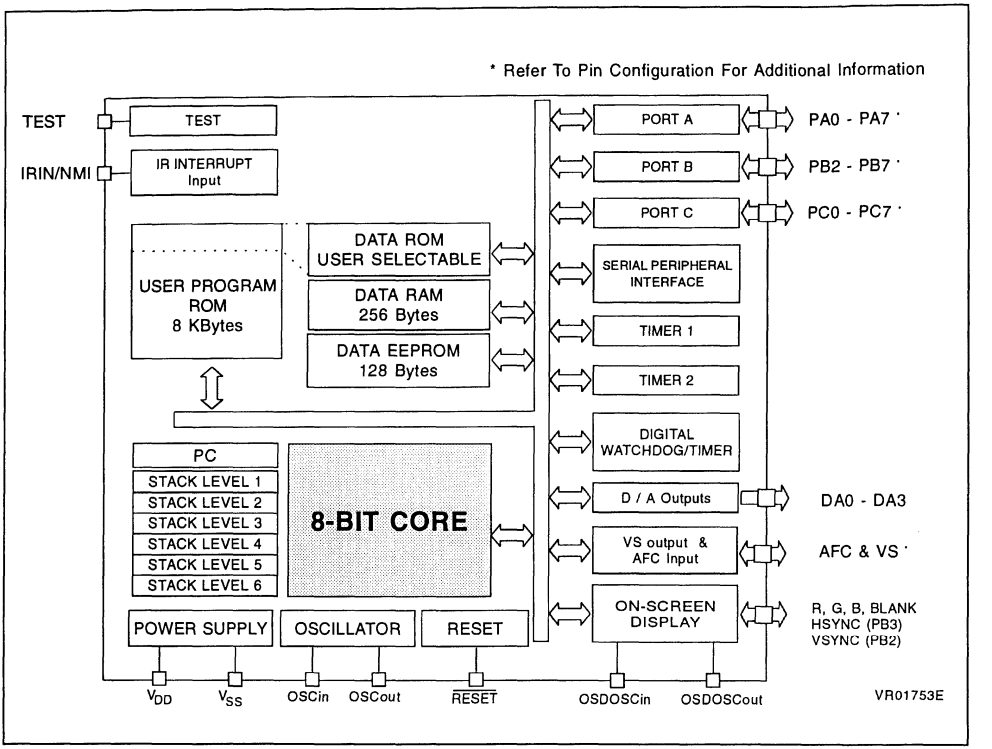


Table 1. Device Summary

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	I/O	KBY I/O	BSW OUT	AFC	VS	D/A	PACK.	EMUL. DEVICES
ST63126	8K	256	128	12	3	4	YES	NO	4	PDIP40	ST63E126
ST63156	8K	256	128	11	3	4	YES	YES	4	PDIP40	ST63E156
ST63140	8K	256	128	6	3	3	YES	YES	1	PDIP28	ST6E140
ST63142	8K	256	128	6	3	3	YES	NO	1	PDIP28	ST63E142

PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCin, OSCout. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to start the microcontroller to the beginning of its program.

TEST. The TEST pin must be held at V_{SS} for normal operation.

PA0-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured as either an input or as an output under software control of the data direction register. Port A has an open-drain (12V drive) output configuration with direct LED driving capability (30mA, 1V).

PB2-PB3, PB5-PB7. These lines are organized as one I/O port (B). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register. PB2-PB3 have a push-pull configuration in output mode while PB5-PB7 are open-drain (5V drive).

PB2 and PB3 lines are connected to the VSYNC and HSYNC control signals of the OSD cell; to provide the right signals to the OSD these I/O lines should be programmed in input mode and the user can read "on the fly" the state of VSYNC and HSYNC signals. PB2 is also connected with the VSYNC Interrupt. The active polarity of VSYNC Interrupt signal is software controlled. The active polarity of these synchronization input pins to the OSD macrocell can be selected by the user as ROM mask option. If the device is specified to have negative logic inputs, then when these signals are low the OSD oscillator stops. If the device is specified to have positive logic inputs, then when these signals are high the OSD oscillator stops.

PB5, PB6 and PB7 lines, when in output modes, are "ANDed" with the SPI control signals. PB5 is connected with the SPI clock signal (SCL), PB6 with the SPI data signal (SDA) while PB7 is connected with SPI enable signal (SEN).

PC0-PC7. These 8 lines are organized as one I/O port (C). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register. PC0-PC2, PC4 have a push-pull configuration in output mode while PC3, PC5-PC7 (OSD signals) are open-drain (5V drive). PC3, PC5 , PC6 and PC7 lines when in output mode are "ANDed" with the character and blank signals of the OSD cell. PC3 is connected with the OSD BLANK signal, PC5, PC6 and PC7 with the OSD R, G and B signals. The active polarity of these signals can be selected by the user as ROM mask option. PC2 is also used as TV set ON-OFF switch (5V drive).

DA0-DA3. These pins are the four PWM D/A outputs (with 32kHz repetition) of the 6-bit on-chip D/A converters. The PWM function can be disabled by software and these lines can be used as general purpose open-drain outputs (12V drive).

IRIN. This pin is the external NMI of the MCU.

OUT1. This pin is the 62.5kHz output specially suited to drive multi-standard chroma processors. This function can be disabled by software and the pin can be used as general purpose open-drain output (12V drive).

BSW0-BSW3. These output pins can be used to select up to 4 tuning bands. These lines are configured as open-drain outputs (12V drive).

KBY0-KBY2. These pins are input only and can be used for keyboard scan. They have CMOS threshold levels with Schmitt Trigger and on-chip 100kΩ pull-up resistors.

AFC. This is the input of the on-chip 10 level comparator that can be used to implement the AFC function. This pin is an high impedance input able to withstand signals with a peak amplitude up to 12V.

OSDOSCin, OSDOSCout. These are the On Screen Display oscillator terminals. An oscillation capacitor and coil network have to be connected to provide the right signal to the OSD.

VS. This is the output pin of the on-chip 14-bit voltage synthesis tuning cell (VS). The tuning signal present at this pin gives an approximate resolution of 40kHz per step over the UHF band. This line is a push-pull output with standard drive (ST63140, ST63156 only).

Table 2. Pin Summary

Pin Function	Description
DA0 to DA3	Output, Open-Drain, 12V
BSW0 to BSW3	Output, Open-Drain, 12V
IRIN	Input, Resistive Bias, Schmitt Trigger
AFC	Input, High Impedance, 12V
OUT1	Output, Open-Drain, 12V
KBY0 to KBY2	Input, Pull-up, Schmitt Trigger
R,G,B, BLANK	Output, Open-Drain, 5V
HSYNC, VSYNC	Input, Pull-up, Schmitt Trigger
OSDOSCin	Input, High Impedance
OSDOSCout	Output, Push-Pull
TEST	Input, Pull-Down
OSCin	Input, Resistive Bias, Schmitt Trigger to Reset Logic Only
OSCout	Output, Push-Pull
RESET	Input, Pull-up, Schmitt Trigger Input
VS	Output, Push-Pull
PA0-PA6	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger, High Drive
PB2-PB3, PB5-PB7	I/O, Push-Pull, 5V, Input Pull-up, Schmitt Trigger
PB5-PB7	I/O, Open-Drain, 5V, Input Pull-up, Schmitt Trigger
PC0-PC2, PC4	I/O, Push-Pull, 5V, Input Pull-up, Schmitt Trigger
PC3, PC5-PC7	I/O, Open-Drain, 5V, Input Pull-up, Schmitt Trigger
V_{DD}, V_{SS}	Power Supply Pins

ST631xx CORE

The Core of the ST631xx Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal addresses, data, and control busses. The in-core communication is arranged as shown in the following block diagram figure; the controller being externally linked to both the reset and the oscillator, while the core is linked to the dedicated on-chip macrocells peripherals via the serial data bus and indirectly for interrupt purposes through the control registers.

Registers

The ST631xx Family Core has six registers and three pairs of flags available to the programmer. They are shown in Figure 5 and are explained in the following paragraphs together with the program and data memory page registers.

Accumulator (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator is addressed in the data space as RAM location at address FFh .
Accordingly, the ST631xx instruction set can use the accumulator as any other register of the data space.

Figure 5. ST631xx Core Programming Model

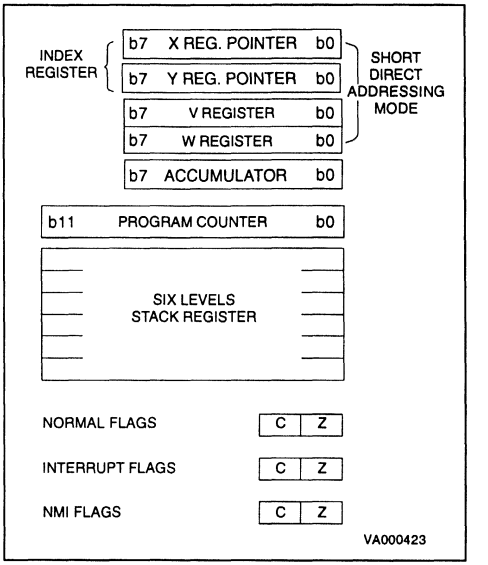
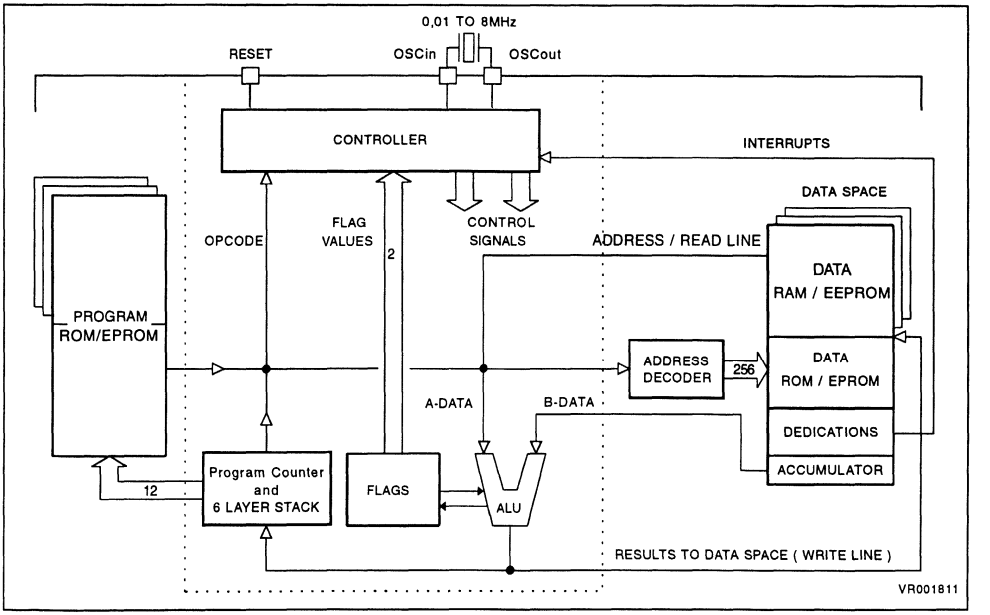


Figure 4. ST631xx Core Block Diagram



ST631xx CORE (Continued)

Indirect Registers (X, Y). These two indirect registers are used as pointers to the memory locations in the data space. They are used in the register-indirect addressing mode.These registers can be addressed in the data space as RAM locations at the 80h (X) and 81h (Y) addresses. They can also be accessed with the direct, short direct, or bit direct addressing modes. Accordingly, the ST631xx instruction set can use the indirect registers as any other register of the data space.

Short Direct Registers (V, W). These two registers are used to save one byte in short direct addressing mode. These registers can be addressed in the data space as RAM locations at the 82h (V) and 83H (W) addresses. They can also be accessed with the direct and bit direct addressing modes. Accordingly, the ST631xx instruction set can use the short direct registers as any other register of the data space.

Program Counter (PC)

The program counter is a 12-bit register that contains the address of the next ROM location to be processed by the core. This ROM location may be an opcode, an operand, or an address of operand. The 12-bit length allows the direct addressing of 4096 bytes in the program space. Nevertheless, if the program space contains more than 4096 locations, the further program space can be addressed by using the Program ROM Page Register. The PC value is incremented, after it is read for the address of the current instruction, by sending it through the ALU, so giving the address of the next byte in the program. To execute relative jumps the PC and the offset values are shifted through the ALU, where they will be added, and the result is shifted back into the PC. The program counter can be changed in the following ways:

JP (Jump) instruction.... PC = Jump address

CALL instruction..... PC = Call address

Relative Branch
instructions..... PC = PC+offset

Interrupt..... PC = Interrupt vector

Reset..... PC = Reset vector

RET & RETI instructions..... PC= Pop (stack)

Normal instruction ..... PC = PC+1

Flags (C, Z)

The ST631xx Core includes three pairs of flags that correspond to 3 different modes: normal mode, interrupt mode and Non-Maskable-Interrupt-Mode. Each pair consists of a CARRY flag and a ZERO flag. One pair (CN, ZN) is used during normal operation, one pair is used during the interrupt mode (CI,ZI) and one is used during the not-maskable interrupt mode (CNMI, ZNMI).

The ST631xx Core uses the pair of flags that corresponds to the actual mode: as soon as an interrupt (resp. a Non-Maskable-Interrupt) is generated, the ST631xx Core uses the interrupt flags (resp. the NMI flags) instead of the normal flags. When the RETI instruction is executed, the normal flags (resp. the interrupt flags) are restored if the MCU was in the normal mode (resp. in the interrupt mode) before the interrupt. Should be observed that each flag set can only be addressed in its own routine (Not-maskable interrupt, normal interrupt or main routine). The interrupt flags are not cleared during the context switching and so, they remain in the state they were at the exit of the last routine switching.

The Carry flag is set when a carry or a borrow occurs during arithmetic operations, otherwise it is cleared. The Carry flag is also set to the value of the bit tested in a bit test instruction, and participates in the rotate left instruction.

The Zero flag is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

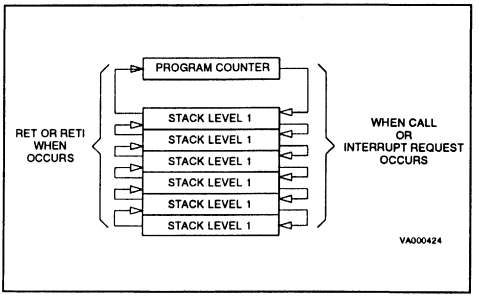
The switching between these three sets is automatically performed when an NMI, an interrupt and a RETI instructions occur. As the NMI mode is automatically selected after the reset of the MCU, the ST631xx Core uses at first the NMI flags.

ST631xxx CORE (Continued)

Stack

The ST631xx Core includes true LIFO hardware stack that eliminates the need for a stack pointer. The stack consists of six separate 12-bit RAM locations that do not belong to the data space RAM area. When a subroutine call (or interrupt request) occurs, the contents of each level is shifted into the next level while the content of the PC is shifted into the first level (the value of the sixth level will be lost). When subroutine or interrupt return occurs (RET or RETI instructions), the first level register is shifted back into the PC and the value of each level is shifted back into the previous level. These two operating modes are described in Figure 6. Since the accumulator, as all other data space registers, is not stored in this stack the handling of this registers shall be performed inside the subroutine. The stack pointer will remain in its deepest position, if more than 6 calls or interrupts are executed, so that the last return address will be lost. It will also remain in its highest position if the stack is empty and a RET or RETI is executed. In this case the next instruction will be executed.

Figure 6. Stack Operation



MEMORY SPACES

The MCUs operate in three different memory spaces: Program Space, Data Space, and Stack Space. A description of these spaces is shown in the following Figures.

Program Space

The program space is physically implemented in the ROM and includes all the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, the reserved test area and user vectors. It is addressed thanks to the 12-bit Program Counter register (PC register) and so, the ST631xx Core can directly address up to 4K bytes of Program Space. Nevertheless, the Program Space can be extended by the addition of 2-Kbyte ROM banks as it is shown in Figure 8 in which the 8K bytes memory is described.

These banks are addressed by pointing to the 000h-7FFh locations of the Program Space thanks to the Program Counter, and by writing the appropriate code in the Program ROM Page Register (PRPR) located at address CAh of the Data Space. Because interrupts and common subroutines should be available all the time only the lower 2K bytes of the 4K program space are bank switched while the upper 2K bytes can be seen as static space. Table 3 gives the codes that allow the selection of the corresponding banks.

Note that, from the memory point of view, the Page 1 and the Static Page represent the same physical memory: it is only a different way of addressing the same location. On the ST631xx a total of 8192 bytes of ROM have been implemented; 7948 are available as user ROM while 244 are reserved for testing.

Figure 8. ST631xx 8K Bytes Program Space Addressing Description

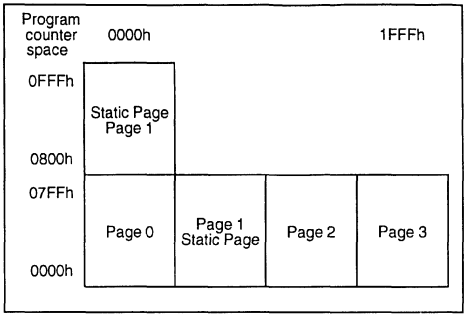
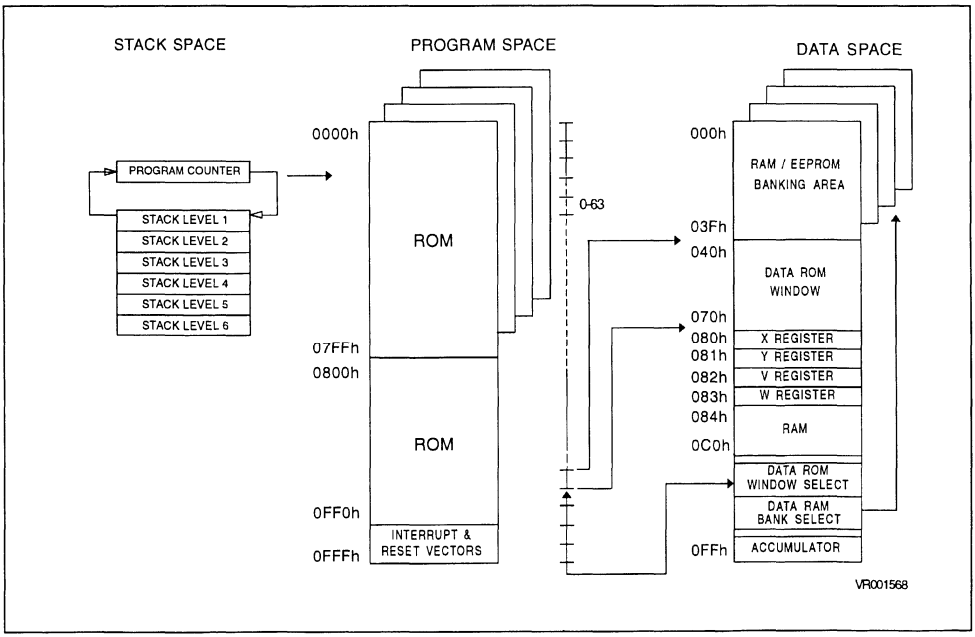
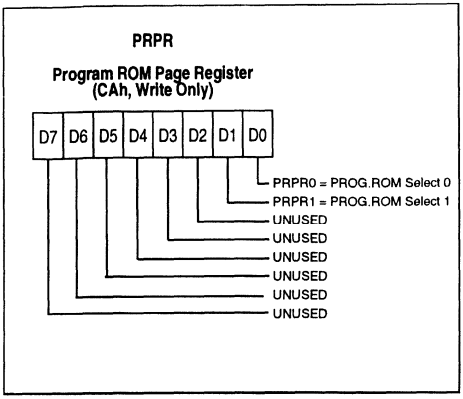


Figure 7. ST631xx Memory Addressing Description Diagram



MEMORY SPACES (Continued)

Figure 11. Program ROM Page Register



D7-D2. These bits are not used.

PRPR1-PRPR0. These are the program ROM banking bits and the value loaded selects the corresponding page to be addressed in the lower part of 4K program address space as specified in Table 3. This register is undefined on reset.

Note:

Only the lower part of address space has been bankswitched because interrupt vectors and common subroutines should be available all the time. The reason of this structure is due to the fact that it is not possible to jump from a dynamic page to another, unless jumping back to the static page, changing contents of PRPR, and, then, jumping to a different dynamic page.

Care is required when handling the PRPR as it is write only. For this reason, it is not allowed to change the PRPR contents while executing interrupts drivers, as the driver cannot save and than restore its previous content. Anyway, this operation may be necessary if the sum of common routines and interrupt drivers will take more than 2K bytes; in this case could be necessary to divide the interrupt driver in a (minor) part in the static page (start and end), and in the second (major) part in one dynamic page. If it is impossible to avoid the writing of this register in interrupts drivers, an image of this register must be saved in a RAM location, and each time the program writes the PRPR it writes also the image register. The image register must be written first, so if an interrupt occurs between the two instructions the PRPR is not affected.

Table 3. Program ROM Page Register Coding

PRPR1	PRPR0	PC11	Memory Page
X	X	1	Static Page (Page1)
0	0	0	Page 0
0	1	0	Page 1 (Static Page)
1	0	0	Page 2
1	1	0	Page 3

This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.

Table 4. ST631xx Program ROM Map

ROM Page	Device Address	Device Address (1)	Description
PAGE 0	0000h-007Fh 0080h-07FFh	0000h-007Fh 0080h-07FFh	Reserved User ROM
PAGE 1 "STATIC"	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
PAGE 2	0000h-000Fh 0010h-07FFh	1000h-100Fh 1010h-17FFh	Reserved User ROM
PAGE 3	0000h-000Fh 0010h-07FFh	1800h-180Fh 1810h-1FFFh	Reserved User ROM

Note 1. EPROM addresses relate to the use of ST63E1xx EPROM Emulation device.

MEMORY SPACES (Continued)

Data Space

The instruction set of the ST631xx Core operates on a specific space, named Data Space that contains all the data necessary for the processing of the program. The Data Space allows the addressing of RAM (256 bytes for the ST631xx family), EEPROM (128 bytes), ST631xx Core/peripheral registers, and read-only data such as constants and the look-up tables.

Figure 12. ST631xx Data Space

b7 b0	
DATA RAM/EEPROM/OSD BANK AREA	000h 03Fh
DATA ROM WINDOW AREA	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
PORT C DATA REGISTER	0C2h
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
PORT C DIRECTION REGISTER	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h
DATA ROM WINDOW REGISTER	0C9h
PROGRAM ROM PAGE REGISTER	0CAh
RESERVED	0CBh
SPI DATA REGISTER	0CCh
RESERVED	0CDh 0D1h
TIMER 1 PRESCALER REGISTER	0D2h
TIMER 1 COUNTER REGISTER	0D3h
TIMER 1 STATUS/CONTROL REG.	0D4h
RESERVED	0D5h 0D7h
WATCHDOG REGISTER	0D8h

Figure 13. ST631xx Data Space (Continued)

b7 b0	
RESERVED	0D9h
TIMER 2 PRESCALER REGISTER	0DAh
TIMER 2 COUNTER REGISTER	0DBh
TIMER 2 STATUS CONTROL REG.	0DCh
RESERVED	0DDh 0DFh
DA0 DATA/CONTROL REGISTER	0E0h
DA1 DATA/CONTROL REGISTER	0E1h
DA2 DATA/CONTROL REGISTER	0E2h
DA3 DATA/CONTROL REGISTER	0E3h
AFC RESULT REGISTER	0E4h
KEYBOARD INPUT REGISTER	0E5h
RESERVED	0E6h
RESERVED	0E7h
DATA RAM BANK REGISTER	0E8h
BSW CONTROL REGISTER	0E9h
EEPROM CONTROL REGISTER	0EAh
SPI CONTROL REGISTER 1	0EBh
SPI CONTROL REGISTER 2	0ECh
VS DATA REGISTER 1	0EDh
VS DATA REGISTER 2	0EEh
OSD CHARAC. BANK SELECT REG.	0EFh
RESERVED	0F0h 0FEh
ACCUMULATOR	0FFh

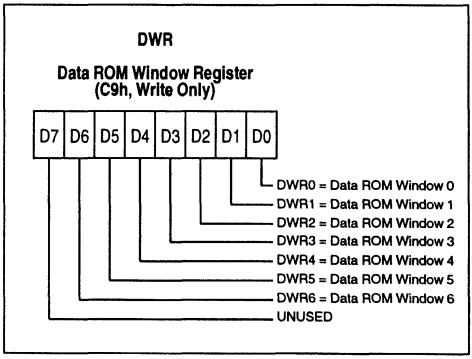
OSD CONTROL REGISTERS LOCATED IN PAGE 6 OF BANKED DATA RAM

VERTICAL START ADDRESS REG.	010h
HORIZONTAL START ADDRESS REG.	011h
VERTICAL SPACE REGISTER	012h
HORIZONTAL SPACE REGISTER	013h
BACKGROUND COLOUR REGISTER	014h
GLOBAL ENABLE REGISTER	017h

MEMORY SPACES (Continued)

Data ROM Addressing. All the read-only data are physically implemented in the ROM in which the Program Space is also implemented. The ROM therefore contains the program to be executed and also the constants and the look-up tables needed for the program. The locations of Data Space in which the different constants and look-up tables are addressed by the ST631xx Core can be considered as being a 64-byte window through which it is possible to access to the read-only data stored in the ROM. This window is located from address 40H to address 7Fh in the Data space and allows the direct reading of the bytes from the address 000h to address 03Fh in the ROM. All the bytes of the ROM can be used to store either instructions or read-only data. Indeed, the window can be moved by step of 64 bytes along the ROM in writing the appropriate code in the Write-only Data ROM Window register (DRWR, location C9h). The effective address of the byte to be read as a data in the ROM is obtained by the concatenation of the 6 less significant bits of the address in the Data Space (as less significant bits) and the content of the DRWR (as most significant bits). So when addressing location 40h of data space, and 0 is loaded in the DRWR, the physical addressed location in ROM is 00h.

Figure 14. Data ROM Window Register



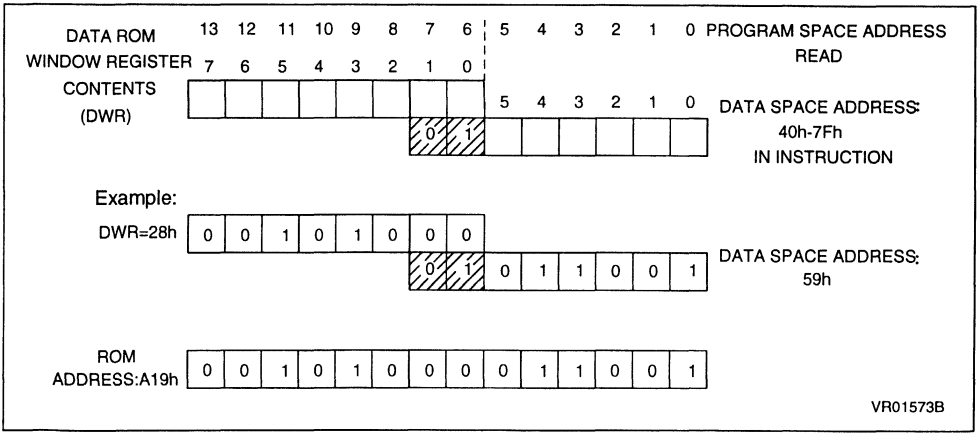
D7. This bit is not used.

DWR6-DWR0. These are the Data Rom Window bits that correspond to the upper bits of data ROM program space. This register is undefined after reset.

This register is undefined on reset. Neither read nor single bit instructions may be used to address this register.

Note. Care is required when handling the DRWR as it is write only. For this reason, it is not allowed to change the DRWR contents while executing interrupts drivers, as the driver cannot save and than restore its previous content. If it is impossible to avoid the writing of this register in interrupts drivers, an image of this register must be saved in a RAM location, and each time the program writes the DRWR it writes also the image register. The image register must be written first, so if an interrupt occurs between the two instructions the DRWR register is not affected.

Figure 15. Data ROM Window Memory Addressing

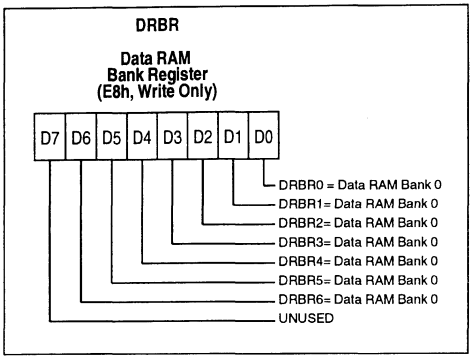


MEMORY SPACES (Continued)

Data RAM/EEPROM/OSD RAM Addressing

In all members of the ST631xx family 64 bytes of data RAM are directly addressable in the data space from 80h to BFh addresses. The additional 192 bytes of RAM, the 128 bytes of EEPROM , and the OSD RAM can be addressed using the banks of 64 bytes located between addresses 00h and 3Fh. The selection of the bank is done by programming the Data RAM Bank Register (DRBR) located at the E8h address of the Data Space. In this way each bank of RAM, EEPROM or OSD RAM can select 64 bytes at a time. No more than one bank should be set at a time.

Figure 16. Data RAM Bank Register



D7. This bit is not used.

DRBR6, DRBR5. Each of these bits, when set, will select one OSD RAM register page.

DRBR4,DRBR3,DRBR2. Each of these bits, when set, will select one RAM page.

DRBR1,DRBR0. These bits select the EEPROM pages.

This register is undefined after reset. Neither read nor single bit instructions may be used to address this register.

Table 5 summarizes how to set the Data RAM Bank Register in order to select the various banks or pages.

Note :
Care is required when handling the DRBR as it is write only. For this reason, it is not allowed to change the DRBR contents while executing interrupts drivers, as the driver cannot save and than restore its previous content. If it is impossible to avoid the writing of this register in interrupts drivers, an image of this register must be saved in a RAM location, and each time the program writes the DRBR it writes also the image register.
The image register must be written first, so if an interrupt occurs between the two instructions the DRBR is not affected.

Table 5. Data RAM Bank Register Set-up

DRBR Value		Selection
Hex.	Binary	
01h	0000 0001	EEPROM Page 0
02h	0000 0010	EEPROM Page 1
04h	0000 0100	RAM Page 2
08h	000 1000	RAM Page 3
10h	0001 0000	RAM Page 4
20h	0010 0000	OSD Page 5
40h	0100 0000	OSD Page 6

MEMORY SPACES (Continued)

EEPROM Description

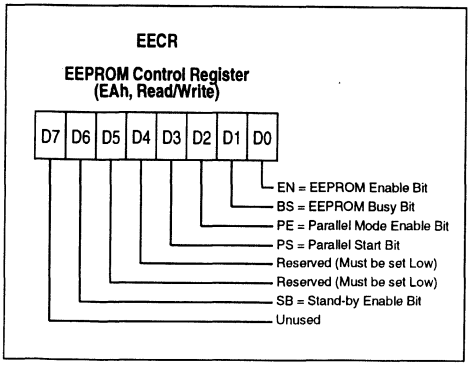
The data space of ST631xx family from 00h to 3Fh is paged as described in Table 5. 128 bytes of EEPROM located in 2 pages of 64 bytes (pages 0, and 1, see Table 5).

Through the programming of the Data RAM Bank Register (DRBR=E8h) the user can select the bank or page leaving unaffected the way to address the static registers. The way to address the "dynamic" page is to set the DRBR as described in Table 5 (e.g. to select EEPROM page 0, the DRBR has to be loaded with content 01h, see Data RAM/EEPROM/OSD RAM addressing for additional information). Bits 0 and 1 of the DRBR are dedicated to the EEPROM.

The EEPROM pages do not require dedicated instructions to be accessed in reading or writing. The EEPROM is controlled by the EEPROM Control Register (EECR=EAh). Any EEPROM location can be read just like any other data location, also in terms of access time.

To write an EEPROM location takes an average time of 5 ms (10ms max) and during this time the EEPROM is not accessible by the Core. A busy flag can be read by the Core to know the EEPROM status before trying any access. In writing the EEPROM can work in two modes: Byte Mode (BMODE) and Parallel Mode (PMODE). The BMODE is the normal way to use the EEPROM and consists in accessing one byte at a time. The PMODE consists in accessing 8 bytes per time.

Figure 17. EEPROM Control Register



D7. Not used

SB. *WRITE ONLY.* If this bit is set the EEPROM is disabled (any access will be meaningless) and the power consumption of the EEPROM is reduced to the leakage values.

D5, D4. Reserved, they must be set to zero.

PS. *SET ONLY.* Once in Parallel Mode, as soon as the user software sets the PS bit the parallel writing of the 8 adjacent registers will start. PS is internally reset at the end of the programming procedure. Note that less than 8 bytes can be written; after parallel programming the remaining undefined bytes will have no particular content.

PE. *WRITE ONLY.* This bit must be set by the user program in order to perform parallel programming (more bytes per time). If PE is set and the "parallel start bit" (PS) is low, up to 8 adjacent bytes can be written at the maximum speed, the content being stored in volatile registers. These 8 adjacent bytes can be considered as row, whose A7, A6, A5, A4, A3 are fixed while A2, A1 and A0 are the changing bytes. PE is automatically reset at the end of any parallel programming procedure. PE can be reset by the user software before starting the programming procedure, leaving unchanged the EEPROM registers.

BS. *READ ONLY.* This bit will be automatically set by the CORE when the user program modifies an EEPROM register. The user program has to test it before any read or write EEPROM operation; any attempt to access the EEPROM while "busy bit" is set will be aborted and the writing procedure in progress completed.

EN. *WRITE ONLY.* This bit MUST be set to one in order to write any EEPROM register. If the user program will attempt to write the EEPROM when EN= "0" the involved registers will be unaffected and the "busy bit" will not be set.

After RESET the content of EECR register will be 00h.

Notes :

When the EEPROM is busy (BS="1") the EECR can not be accessed in write mode, it is only possible to read BS status. This implies that as long as the EEPROM is busy it is not possible to change the status of the EEPROM control register. EECR bits 4 and 5 are reserved for test purposes, and must never be set to "1".

MEMORY SPACES (Continued)

Additional Notes on Parallel Mode. If the user wants to perform a parallel programming the first action should be the set to one the PE bit; from this moment the first time the EEPROM will be addressed in writing, the ROW address will be latched and it will be possible to change it only at the end of the programming procedure or by resetting PE without programming the EEPROM. After the ROW address latching the Core can "see" just one EEPROM row (the selected one) and any attempt to write or read other rows will produce errors. Do not read the EEPROM while PE is set.

As soon as PE bit is set, the 8 volatile ROW latches are cleared. From this moment the user can load data in the whole ROW or just in a subset. PS setting will modify the EEPROM registers corresponding to the ROW latches accessed after PE. For example, if the software sets PE and accesses EEPROM in writing at addresses 18h,1Ah,1Bh and then sets PS, these three registers will be modified at the same time; the remaining bytes will have no particular content. Note that PE is internally reset at the end of the programming procedure. This implies that the user must set PE bit between two parallel programming procedures. Anyway the user can set and then reset PE without performing any EEPROM programming. PS is a set only bit and is internally reset at the end of the programming procedure. Note that if the user tries to set PS while PE is not set there will not be any programming procedure and the PS bit will be unaffected. Consequently PS bit can not be set if EN is low. PS can be affected by the user set if, and only if, EN and PE bits are also set to one.

STACK SPACE

The stack space consists of six 12 bit registers that are used for stacking subroutine and interrupt return addresses plus the current program counter register.

INTERRUPT

The ST631xx Core can manage 4 different maskable interrupt sources, plus one non-maskable interrupt source (top priority level interrupt). Each source is associated with a particular interrupt vector that contains a Jump instruction to the related interrupt service routine. Each vector is located in the Program Space at a particular address (see Table 6). When a source provides an interrupt request, and the request processing is also enabled by the ST631xx Core, then the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction). Finally, the PC is loaded with the address of the Jump instruction and the interrupt routine is processed.

The relationship between vector and source and the associated priority is hardware fixed for the different ST631xx devices. For some interrupt sources it is also possible to select by software the kind of event that will generate the interrupt.

All interrupts can be disabled by writing to the GEN bit (global interrupt enable) of the interrupt option register (address C8h). After a reset, ST631xx is in non maskable interrupt mode, so no interrupts will be accepted and NMI flags will be used, until a RETI instruction is executed. If an interrupt is executed, one special cycle is made by the core, during that the PC is set to the related interrupt vector address. A jump instruction at this address has to redirect program execution to the beginning of the related interrupt routine. The interrupt detecting cycle, also resets the related interrupt flag (not available to the user), so that another interrupt can be stored for this current vector, while its driver is under execution.

If additional interrupts arrive from the same source, they will be lost. NMI can interrupt other interrupt routines at any time, while other interrupts cannot interrupt each other. If more than one interrupt is waiting for service, they are executed according to their priority. The lower the number, the higher the priority. Priority is, therefore, fixed. Interrupts are checked during the last cycle of an instruction (RETI included). Level sensitive interrupts have to be valid during this period.

INTERRUPT (Continued)

Table 6. Interrupt Vectors/Sources Relationships

Interrupt Source	Associated Vector	Vector Address
IRIN/NMI Pin (1)	Interrupt Vector # 0 (NMI)	0FFCh-0FFDh
None (2)	Interrupt Vector # 1	0FF6h-0FF7h
Vsync	Interrupt Vector # 2	0FF4h-0FF5h
Timer 1	Interrupt Vector # 3	0FF2h-0FF3h
Timer 2	Interrupt Vector # 4	0FF0h-0FF1h

Notes:

1. This pin is associated with the NMI Interrupt Vector
2. This vector is not used in ST631xx.

Interrupt Vectors/Sources

The ST631xx Core includes 5 different interrupt vectors in order to branch to 5 different interrupt routines. The interrupt vectors are located in the fixed (or static) page of the Program Space.

The interrupt vector associated with the non-maskable interrupt source is named interrupt vector #0. It is located at addresses FFCh,FFDh in the Program Space. This vector is associated with the PC6/IRIN pin.

The interrupt vectors located at addresses (FF6h,FF7h), (FF4h,FF5h), (FF2h,FF3h), (FF0h,FF1h) are named interrupt vectors #1, #2, #3 and #4 respectively. These vectors are associated with TIMER 2 (#4), VSYNC (#2), and TIMER 1 (#3). Interrupt vector (#1) is not used on ST631xx.

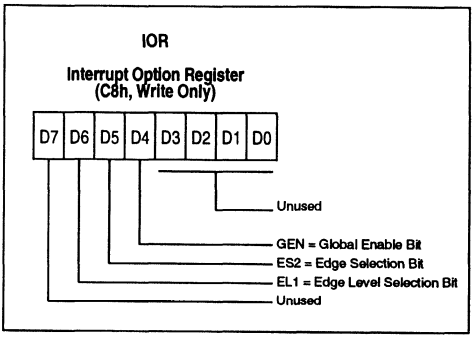
Interrupt Priority

The non-maskable interrupt request has the highest priority and can interrupt any other interrupt routines at any time, nevertheless the other interrupts cannot interrupt each other. If more than one interrupt request is pending, they are processed by the ST631xx Core according to their priority level: vector #1 has the higher priority while vector #4 the lower. The priority of each interrupt source is hardware fixed.

Interrupt Option Register

The Interrupt Option Register (IOR register, location C8h) is used to enable/disable the individual interrupt sources and to select the operating mode of the external interrupt inputs. This register can be addressed in the Data Space as RAM location at the C8h address, nevertheless it is write-only register that can not be accessed with single-bit operations. The operating modes of the external interrupt inputs associated to interrupt vectors #1 and #2 are selected through bits 5 and 6 of the IOR register.

Figure 18. Interrupt Option Register



D7. Not used.

EL1. This is the Edge/Level selection bit of interrupt #1. When set to one, the interrupt is generated on low level of the related signal; when cleared to zero, the interrupt is generated on falling edge. The bit is cleared to zero after reset and as no interrupt source is associated to vector #1 on ST631xx, the user must keep this bit at zero to avoid ghost interrupts from this source.

ES2. This is the edge selection bit on interrupt #2. This bit is used on the ST631xx devices with on-chip OSD generator for VSYNC detection.

GEN. This is the global enable bit. When set to one all interrupts are globally enabled; when this bit is cleared to zero all interrupts are disabled (EXcluding NMI).

D3 - D0. These bits are not used.

INTERRUPT (Continued)

Interrupt Procedure. The interrupt procedure is very similar to a call procedure, indeed the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event the user does not know about the context and the time at which it occurred. As a result the user should save all the data space registers which will be used inside the interrupt routines. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes which are automatically switched and so these do not need to be saved.

The following list summarizes the interrupt procedure:

ST631xx actions

- Interrupt detection
- The flags C and Z of the main routine are exchanged with the flags C and Z of the interrupt routine (or the NMI flags)
- The value of the PC is stored in the first level of the stack
- The normal interrupt lines are inhibited (NMI still active)
- First internal latch is cleared
- The related interrupt vector is loaded in the PC.

User actions

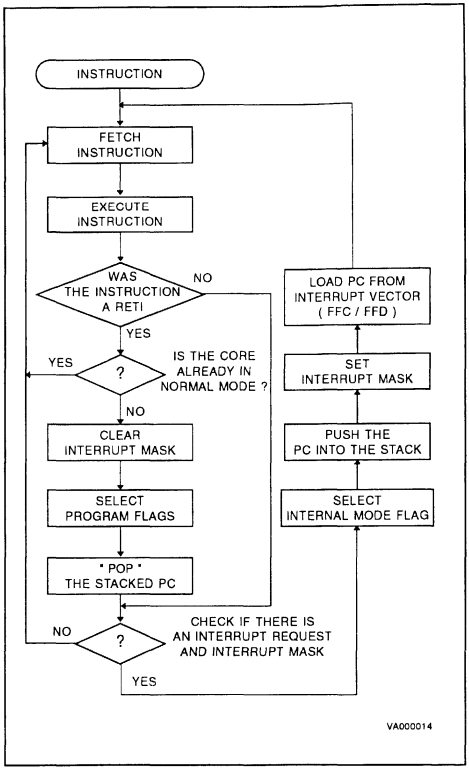
- User selected registers are saved inside the interrupt service routine (normally on a software stack)
- The source of the interrupt is found by polling (if more than one source is associated to the same vector) the interrupt flag of the source.
- Interrupt servicing
- Return from interrupt (RETI)

ST631xx actions

- Automatically the ST631xx core switches back to the normal flags (or the interrupt flags) and pops the previous PC value from the stack

The interrupt routine begins usually by the identification of the device that has generated the interrupt request (by polling). The user should save the registers which are used inside the interrupt routine (that holds relevant data) into a software stack. After the RETI instruction execution, the core carries out the previous actions and the main routine can continue.

Figure 19. Interrupt Processing Flow-Chart



ST631xx Interrupt Details

IR Interrupt (#0). The IRIN Interrupt is connected to the first interrupt #0 (NMI, 0FFCh). If enabled, then an interrupt will be generated on a rising edge at the pin.

Interrupt (#1). On ST631xx no sources are associated to vector (#1). To avoid any ghost interrupt due to interrupt (#1) the user must keep the EL1 bit of IOR register to zero.

INTERRUPT (Continued)

VSYNC Interrupt (#2). The VSYNC Interrupt is connected to the interrupt #2. When disabled the VSYNC INT signal is low. Bit 5 of the interrupt option register C8h is used to select the negative edge (ES2=0) or the positive edge (ES2=1); the edge will depend on the application. Note that once an edge has been latched, then the only way to remove the latched signal is to service the interrupt. Care must be taken not to generate spurious interrupts. This interrupt may be used for synchronize to the VSYNC signal in order to change characters in the OSD only when the screen is on vertical blanking (if desired). This method may also be used to blink characters.

TIMER 1 Interrupt (#3). The TIMER 1 Interrupt is connected to the fourth interrupt #3 (0FF2h) which detects a low level (latched in the timer).

TIMER 2 Interrupt (#4). The TIMER 2 Interrupt is connected to the fifth interrupt #4 (0FF0h) which detects a high to low level (latched in the timer).

Notes: Global disable does not reset edge sensitive interrupt flags. These edge sensitive interrupts become pending again when global disabling is released. Moreover, edge sensitive interrupts are stored in the related flags also when interrupts are globally disabled, unless each edge sensitive interrupt is also individually disabled before the interrupting event happens. Global disable is done by clearing the GEN bit of Interrupt option register, while any individual disable is done in the control register of the peripheral. The on-chip Timer peripherals have an interrupt request flag bit (TMZ), this bit is set to one when the device wants to generate an interrupt request and a mask bit (ETI) that must be set to one to allow the transfer of the flag bit to the Core.

RESET

The ST631xx device can be reset in two ways: by the external reset input (RESET) tied low, by power-on reset and by the digital Watchdog peripheral.

RESET Input

The external active low RESET pin is used to reset the ST631xx devices and provide an orderly software startup procedure. The activation of the RESET pin may occur in the RUN or WAIT mode. Even short pulses at the reset pin will be accepted since the reset signal is latched internally and is only cleared after 2048 clocks at the oscillator pin. The clocks from the oscillator pin to the reset circuitry are buffered by a Schmitt Trigger so that an oscillator in start-up conditions will not give spurious clocks. The MCU is configured in the Reset mode as long as the signal of the RESET pin is low. The processing of the program is stopped and the standard Input/Output ports (port A, port B and port C) are in the input state (except PC2). As soon as the level on the RESET pin becomes high, the initialization sequence is executed.

Watchdog Reset

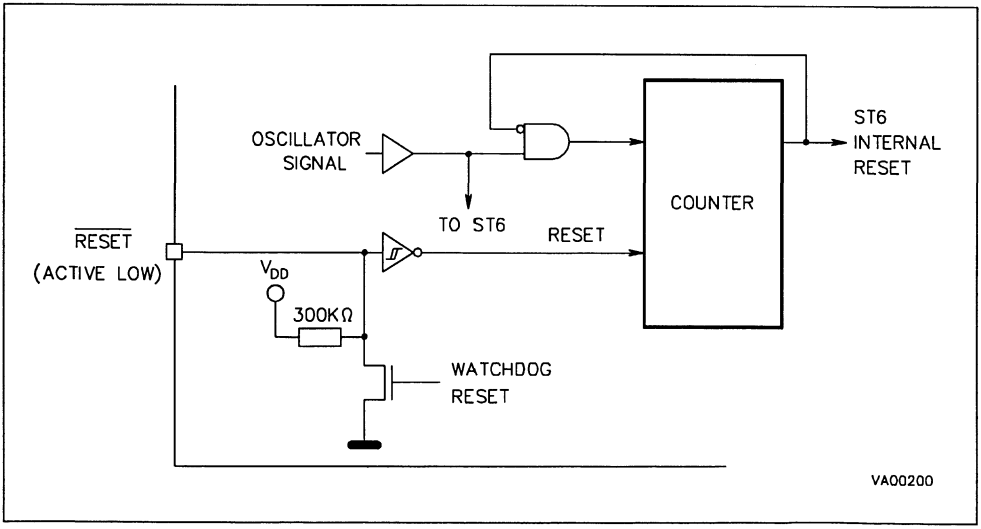
The ST631xx devices are provided with an on-chip hardware activated digital watchdog function in order to provide a graceful recovery from a software upset. If the watchdog register is not refreshed and the end-of-count is reached, then the reset state will be latched into the MCU and an internal circuit pulls down the RESET pin. This also resets the watchdog which subsequently turns off the pull-down and activates the pull-up device at the RESET pin. This causes the positive transition at the RESET pin. The MCU will then exit the reset state after 2048 clocks on the oscillator pin.

Application Notes

An external resistor between V_{DD} and reset pin is not required because an internal pull-up device is provided. The user may prefer to add an external pull-up resistor.

An internal Power-on device does not guarantee that the MCU will exit the reset state when V_{DD} is above 4.5V and therefore the RESET pin should be externally controlled.

Figure 20. Internal Reset Circuit



RESET (Continued)

Figure 21. Reset & Interrupt Processing Flow-Chart

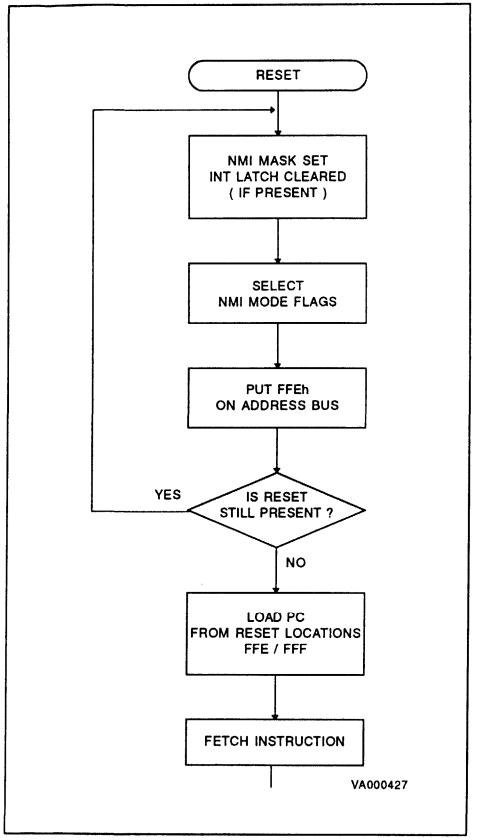
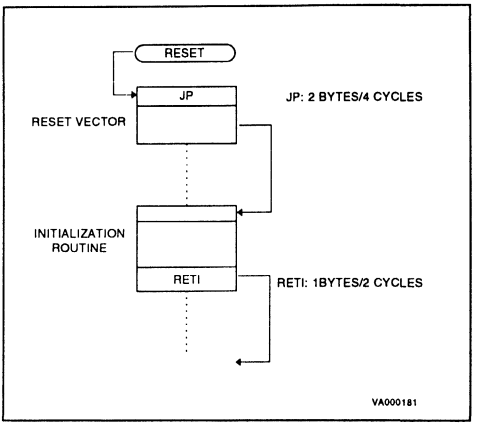


Figure 22. Restart Initialization Program Flow-Chart



MCU Initialization Sequence

When a reset occurs the stack is reset to the program counter, the PC is loaded with the address of the reset vector (located in the program ROM at addresses FFEh & FFFh). A jump instruction to the beginning of the program has to be written into these locations. After a reset a NMI is automatically activated so that the core is in non-maskable interrupt mode to prevent false or ghost interrupts during the restart phase. Therefore the restart routine should be terminated by a RETI instruction to switch to normal mode and enable interrupts. If no pending interrupt is present at the end of the reset routine the ST631xx will continue with the instruction after the RETI; otherwise the pending interrupt will be serviced.

WAIT & STOP MODES

The STOP and WAIT modes have been implemented in the ST631xx Core in order to reduce the consumption of the device when the latter has no instruction to execute. These two modes are described in the following paragraphs. On ST631xx as the hardware activated digital watchdog function is present the STOP instruction is de-activated and any attempt to execute it will cause the automatic execution of a WAIT instruction.

WAIT Mode

The configuration of the MCU in the WAIT mode occurs as soon as the WAIT instruction is executed. The microcontroller can also be considered as being in a "software frozen" state where the Core stops processing the instructions of the routine, the contents of the RAM locations and peripheral registers are saved as long as the power supply voltage is higher than the RAM retention voltage but where the peripherals are still working. The WAIT mode is used when the user wants to reduce the consumption of the MCU when it is in idle, while not losing count of time or monitoring of external events. The oscillator is not stopped in order to provide clock signal to the peripherals. The timers counting may be enabled (writing the PSI bit in TSCR register) and the timer interrupt may be also enabled before entering the WAIT mode; this allows the WAIT mode to be left when timer interrupt occurs. If the exit from the WAIT mode is performed with a general RESET (either from the activation of the external pin or by watchdog reset) the MCU will enter a normal reset procedure as described in the RESET chapter. If an interrupt is generated during WAIT mode the MCU behaviour depends on the state of the ST631xx Core before the initialization of the WAIT sequence, but also of the kind of the interrupt request that is generated. This case will be described in the following paragraphs. In any case, the ST631xx Core does not generate any delay after the occurrence of the interrupt because the oscillator clock is still available.

STOP Mode

On ST631xx the hardware watchdog is present and the STOP instruction has been de-activated. Any attempt to execute a STOP will cause the automatic execution of a WAIT instruction.

Exit from WAIT Mode

The following paragraphs describe the output procedure of the ST631xx Core from WAIT mode when an interrupt occurs. It must be noted that the restart sequence depends on the original state of the MCU (normal, interrupt or non-maskable interrupt mode) before the start of the WAIT sequence, but also of the type of the interrupt request that is generated. In all cases the GEN bit of IOR has to be set to 1 in order to restart from WAIT Mode. Contrary to the operation of NMI in the RUN Mode, the NMI is masked in WAIT Mode if GEN=0.

Normal Mode. If the ST631xx Core was in the main routine when the WAIT instruction has been executed, the ST631xx Core outputs from the wait mode as soon as any interrupt occurs; the related interrupt routine is executed and at the end of the interrupt service routine the instruction that follows the WAIT instruction is executed if no other interrupts are pending.

Non-maskable Interrupt Mode. If the WAIT instruction has been executed during the execution of the non-maskable interrupt routine, the ST631xx Core outputs from the wait mode as soon as any interrupt occurs: the instruction that follows the WAIT instruction is executed and the ST631xx Core is still in the non-maskable interrupt mode even if another interrupt has been generated.

Normal Interrupt Mode. If the ST631xx Core was in the interrupt mode before the initialization of the WAIT sequence, it outputs from the wait mode as soon as any interrupt occurs. Nevertheless, two cases have to be considered:

- If the interrupt is a normal interrupt, the interrupt routine in which the WAIT was entered will be completed with the execution of the instruction that follows the WAIT and the ST631xx Core is still in the interrupt mode. At the end of this routine pending interrupts will be serviced in accordance to their priority.
- If the interrupt is a non-maskable interrupt, the non-maskable routine is processed at first. Then, the routine in which the WAIT was entered will be completed with the execution of the instruction that follows the WAIT and the ST631xx Core is still in the normal interrupt mode.

Notes :
If all the interrupt sources are disabled, the restart of the MCU can only be done by a Reset activation. The Wait instruction is not executed if an enabled interrupt request is pending. In the ST631xx the hardware activated digital watchdog function is present. As the watchdog is always activated the STOP instruction is de-activated and any attempt to execute the STOP instruction will cause an execution of a WAIT instruction.

ON-CHIP CLOCK OSCILLATOR

The internal oscillator circuit is designed to require a minimum of external components. A crystal quartz, a ceramic resonator, or an external signal (provided to the OSCin pin) may be used to generate a system clock with various stability/cost tradeoffs. The typical clock frequency is 8MHz. Please note that different frequencies will affect the operation of those peripherals (D/As, SPI, 62.5 kHz OUT) whose reference frequencies are derived from the system clock.

The different clock generator options connection methods are shown in Figures 24 and 25. One machine cycle takes 13 oscillator pulses; 12 clock pulses are needed to increment the PC while and additional 13th pulse is needed to stabilize the internal latches during memory addressing. This means that with a clock frequency of 8MHz the machine cycle is 1.625µSec.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially RS), oscillator load capacitance (CL), IC parameters, ambient temperature, and supply voltage.It must be observed that the crystal or ceramic leads and circuit connections must be as short as possible. Typical values for CL1 and CL2 are in the range of 15pF to 22pF but these should be chosen based on the crystal manufacturers specification. Typical input capacitance for OSCin and OSCout pins is 5pF.

The oscillator output frequency is internally divided by 13 to produce the machine cycle and by 12 to produce the Timer and the Watchdog clock. A byte cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five byte cycles to be executed (See Table 7).

Table 7. Intructions Timing with 8MHz Clock

Instruction Type	Cycles	Execution Time
Branch if set/reset	5 Cycles	8.125µs
Branch & Subroutine Branch	4 Cycles	6.50µs
Bit Manipulation	4 Cycles	6.50µs
Load Instruction	4 Cycles	6.50µs
Arithmetic & Logic	4 Cycles	6.50µs
Conditional Branch	2 Cycles	3.25µs
Program Control	2 Cycles	3.25µs

Figure 23. Clock Generator Option (1)

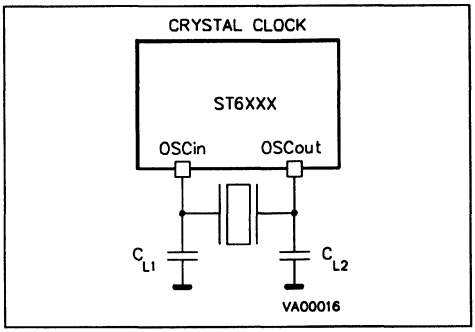


Figure 24. Clock Generator Option (2)

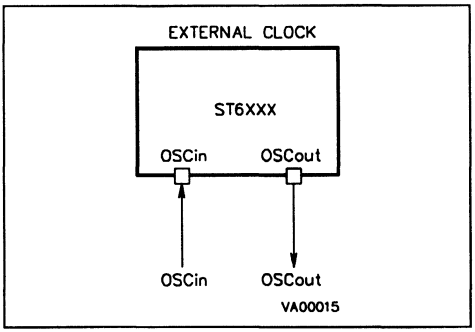
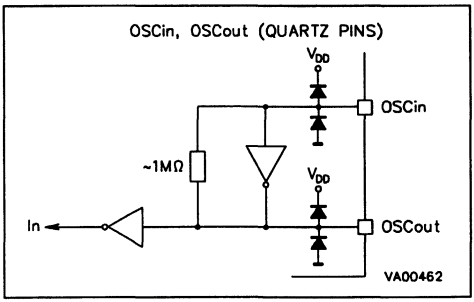


Figure 25. OSCin, OSCout Diagram



INPUT/OUTPUT PORTS

The ST631xx microcontrollers use three standard I/O ports (A,B,C) with up to eight pins on each port; refer to the device pin configurations to see which pins are available.

Each line can be individually programmed either in the input mode or the output mode as follows by software.

- Output
- Input with on-chip pull-up resistor (selected by software)
- Input without on-chip pull-up resistor (selected by software)

Note: pins with 12V open-drain capability do not have pull-up resistors.

In output mode the following hardware configurations are available:

- Open-drain output 12V (PA0-PA7)
- Open-drain output 5V (PB5-PB7, PC3, PC5-PC7)
- Push-pull output (PB0-PB4, PC0-PC2, PC4)

The lines are organized in three ports (port A,B,C). The ports occupy 6 registers in the data space. Each bit of these registers is associated with a particular line (for instance, the bits 0 of the Port A Data and Direction registers are associated with the PA0 line of Port A).

There are three Data registers (DRA, DRB, DRC), that are used to read the voltage level values of the lines programmed in the input mode, or to write the logic value of the signal to be output on the lines configured in the output mode. The port Data Registers can be read to get the effective logic levels of the pins, but they can be also written by the user software, in conjunction with the related Data Direction Register, to select the different input mode options. Single-bit operations on I/O registers (bit set/reset instructions) are possible but care is necessary because reading in input mode is made from I/O pins and therefore might be influenced by the external load, while writing will directly affect the Port data register causing an undesired changes of the input configuration. The three Data Direction registers (DDRA, DDRB, DDRC) allow the selection of the direction of each pin (input or output).

All the I/O registers can be read or written as any other RAM location of the data space. During the initialization of the MCU, all the I/O registers are cleared and the input mode with pull-up is selected on all the pins thus avoiding pin conflicts (with the exception of PC2 which is set in output mode and is set low).

Details of I/O Ports

When programmed as an input a pull-up resistor (if available) can be switched active under program control. When programmed as an output the I/O port will operate either in the push-pull mode or the open-drain mode according to the hardware fixed configuration as specified below.

Port A. PA0-PA7 are available as an open-drain only (no push-pull programmability and no resistive pull-up in input mode) capable of withstanding 12V while the normal open drain has standard ratings of V_{DD} + 0.3V. This I/O port has been specially designed for direct LED driving and is able to sink up to 30mA with a maximum V_{OL} of 1V.

Some Port B and C lines are also used as I/O buffers for signals coming from the on-chip SPI and OSD.

In this case the final signal on the output pin is equivalent to a wired AND with the programmed data output.

If the user needs to use the SPI or the OSD, then the I/O line should be set in output mode while the open-drain configuration is fixed in hardware ; the corresponding data bit must be set to one.

PB2 and PB3 must be programmed in input mode to provide the HSYNC and VSYNC input signals to the OSD.

On ST631xx the I/O pins with double or special functions are:

- PB2/VSYNC (connected to the OSD VSYNC signal)
- PB3/HSYNC (connected to the OSD HSYNC signal)
- PB5/SCL (connected to the SPI clock signal)
- PB6/SDA (connected to the SPI data signal)
- PB7/SEN (connected to the SPI enable signal)
- PC2-ON-OFF, this I/O is specially suited to TV SET ON-OFF and for this reason at reset the related Data Direction bit will be automatically set to one (I/O line is in output mode), while the rest of the port is in input mode
- PC3/BLANK (connected to the OSD Blank signal)
- PC5/R, PC6/G, PC7/B (connected to the OSD R-G-B signals)

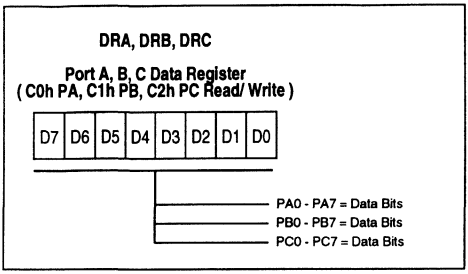
All the Port A,B and C I/O lines have Schmitt-trigger input configuration with a typical hysteresis of 1V.

INPUT/OUTPUT PORTS (Continued)

I/O Pin Programming

Each pin can be individually programmed as input or output with different input and output configurations. This is achieved by writing to the relevant bit in the data (DR) and data direction register (DDR). Table 8 shows all the port configurations that can be selected by the user software.

Figure 24. Port A, B, C Data Register



PA7-PA0. These are the I/O port A data bits. Reset at power-on.

PB7-PB0. These are the I/O port B data bits. Reset at power-on.

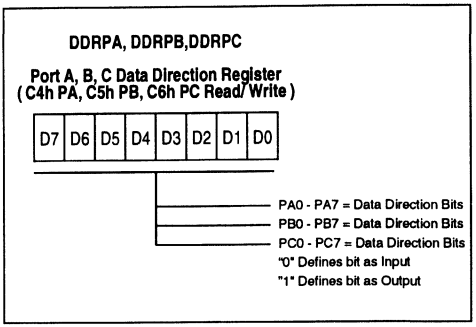
PC7-PC0. These are the I/O port C data bits. Reset at power-on.

Table 8. I/O Port Options Selection

DDR	DR	Mode	Option
0	0	Input	With on-chip pull-up resistor
0	1	Input	Without on-chip pull-up resistor
1	X	Output	Open-drain or Push-Pull

Note: X: Means don't care.

Figure 25. Port A, B, C Data Direction Register



PA7-PA0. These are the I/O port A data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Reset at power-on.

PB7-PB0. These are the I/O port B data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Reset at power-on.

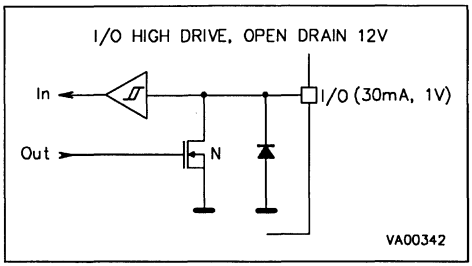
PC7-PC0. These are the I/O port C data direction bits. When a bit is cleared to zero the related I/O line is in input mode, if bit is set to one the related I/O line is in output mode. Set to 04h at power-on. Bit 2 (PC2 pin) is set to one (output mode selected) as this line is intended for TV ON-OFF switching.

INPUT/OUTPUT PORTS (Continued)

Input/Output Configurations

The following schematics show the I/O lines hardware configuration for the different options. Figure 30 shows the I/O configuration for an I/O pin with open-drain 12V capability (standard drive and high drive). Figure 31 shows the I/O configuration for an I/O pin with push-pull and with open drain 5V capability.

Figure 26. I/O Configuration Diagram (Open Drain 12V)

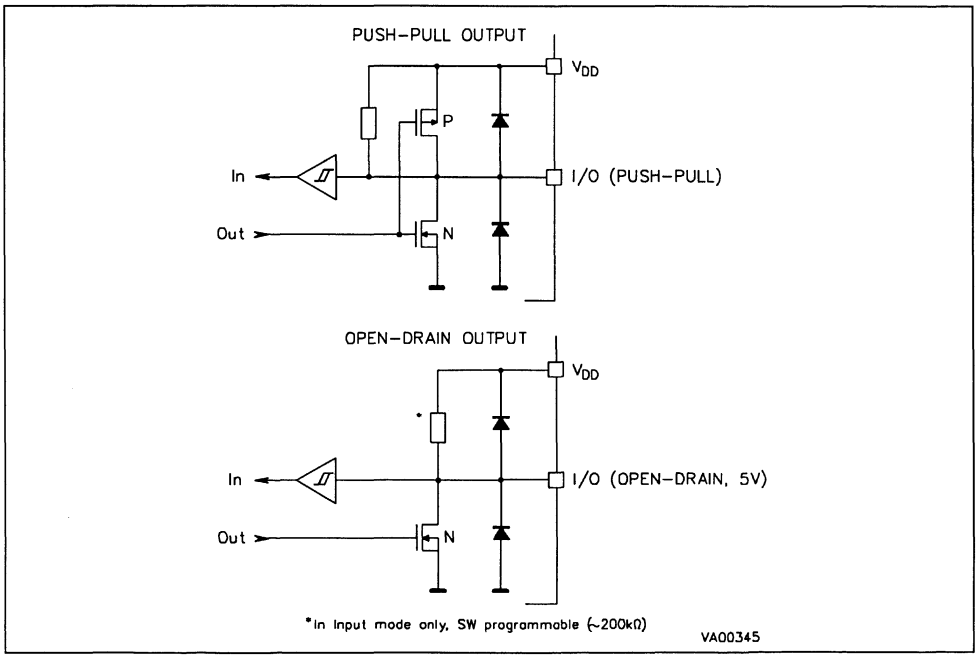


Notes :

The WAIT instruction allows the ST631xx to be used in situations where low power consumption is needed. This can only be achieved however if the I/O pins either are programmed as inputs with well defined logic levels or have no power consuming resistive loads in output mode. As the same die is used for the different ST631xx versions the unavailable I/O lines of ST631xx should be programmed in output mode.

Single-bit operations on I/O registers are possible but **care is necessary** because reading in input mode is made from I/O pins while writing will directly affect the Port data register causing an undesired changes of the input configuration.

Figure 27. I/O Configuration Diagram (Open Drain 5V, Push-pull)



TIMERS

The ST631xx devices offer two on-chip Timer peripherals consisting of an 8-bit counter with a 7-bit programmable prescaler, thus giving a maximum count of 2^{15}, and a control logic that allows configuring the peripheral operating mode. Figure 30 shows the timer block diagram. The content of the 8-bit counters can be read/written in the Timer/Counter registers TCR that can be addressed in the data space as RAM location at addresses D3h (Timer 1) and DBh (Timer 2). The state of the 7-bit prescaler can be read in the PSC register at addresses D2h (Timer 1) and DAh (Timer 2). The control logic is managed by TSCR registers at D4h (Timer 1) and DCh (Timer 2) addresses as described in the following paragraphs.

The following description applies to both Timer 1 and Timer 2. The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (timer zero) bit in the TSCR is set to one. If the ETI (enable timer interrupt) bit in the TSCR is also set to one an interrupt request, associated to interrupt vector #3 (for Timer 1) and #1 for Timer 2, is generated. The interrupt of the timer can be used to exit the MCU from the WAIT mode.

The prescaler decrements on rising edge. The prescaler input is the oscillator frequency divided by 12 or an external clock at TIMER pin (this is not available in ST631xx).

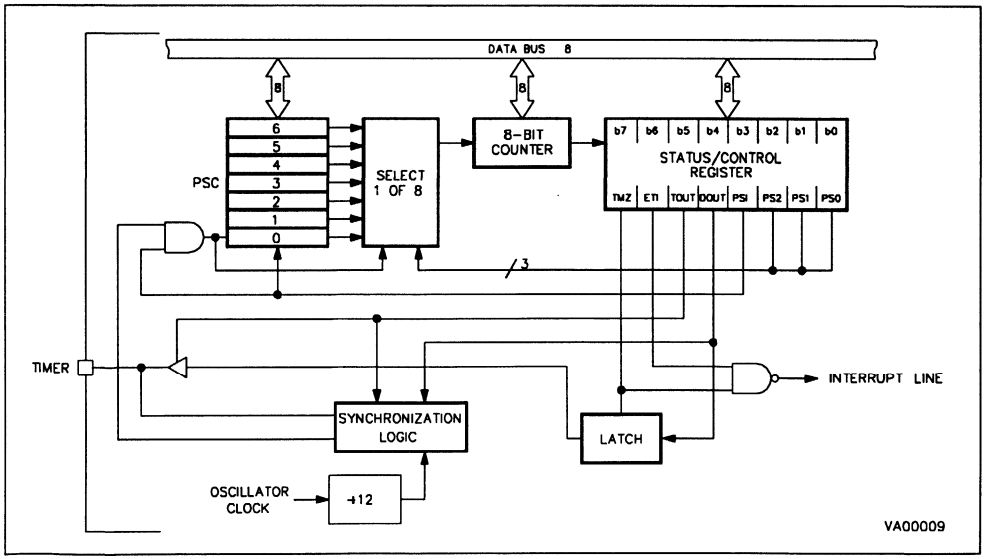
Depending on the division factor programmed by PS2/PS1/PS0 (see table 9) bits in the TSCR, the clock input of the timer/counter register is multiplexed to different sources.

On division factor 1, the clock input of the prescaler is also that of timer/counter; on factor 2, bit 0 of prescaler register is connected to the clock input of TCR.

This bit changes its state with the half frequency of prescaler clock input. On factor 4, bit 1 of PSC is connected to clock input of TCR, and so on. On division factor 128, the MSB bit 6 of PSC is connected to clock input of TCR. The prescaler initialize bit (PSI) in the TSCR register must be set to one to allow the prescaler (and hence the counter) to start. If it is cleared to zero then all of the prescaler bits are set to one and the counter is inhibited from counting.

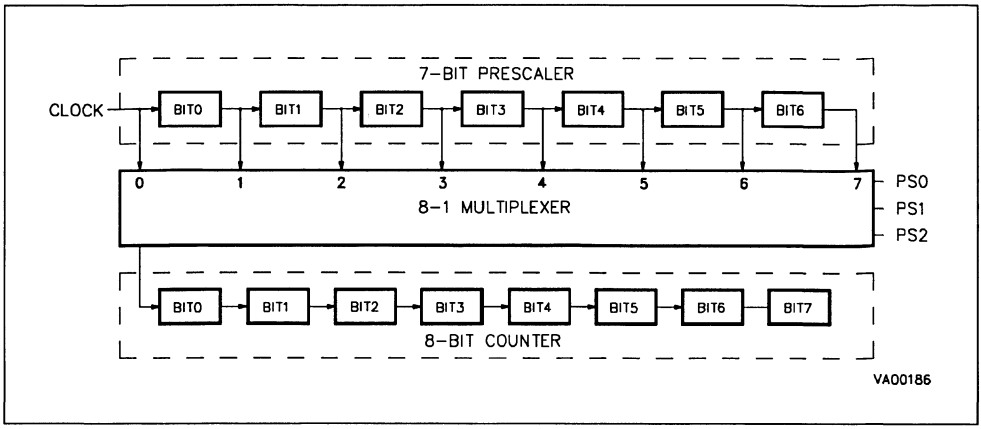
The prescaler can be given any value between 0 and 7Fh by writing to the related register address, if bit PSI in the TSCR register is set to one. The tap of the prescaler is selected using the PS2/PS1/PS0 bits in the control register. Figure 33 shows the timer working principle.

Figure 28. Timer Peripheral Block Diagram



TIMERS (Continued)

Figure 29. Timer Working Principle



Timer Operating Modes

As the external TIMER pin is not available on ST631xx devices, the only allowed operating mode is the output mode that have to be selected by setting to 1 bit 4 and by clearing to 0 bit 5 in the TSCR1 register. This procedure will enable both Timer 1 and Timer 2.

Output Mode (TSCR1 D4 = 1, TSCR1 D5 = 0). On this mode the timer prescaler is clocked by the prescaler clock input (OSC/12). The user can select the desired prescaler division ratio through the PS2/PS1/PS0 bits. When TCR count reaches 0, it sets the TMZ bit in the TSCR.

The TMZ bit can be tested under program control to perform timer functions whenever it goes high. Bit D4 and D5 on TSCR2 (Timer 2) register are not implemented.

Timer Interrupt

When the counter register decrements to zero and the software controlled ETI (enable timer interrupt) bit is set to one then an interrupt request associated to interrupt vector #3 (for Timer 1) and to interrupt vector #4 (for Timer 2) is generated. When the counter decrements to zero also the TMZ bit in the TSCR register is set to one.

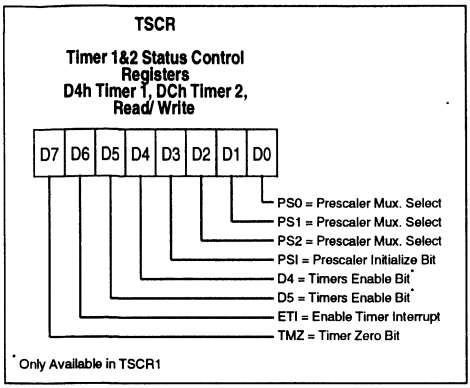
Notes :

TMZ is set when the counter reaches 00h ; however, it may be set by writing 00h in the TCR register or setting the bit 7 of the TSCR register. TMZ bit must be cleared by user software when servicing the timer interrupt to avoid undesired interrupts when leaving the interrupt service routine. After reset, the 8-bit counter register is loaded to FFh while the 7-bit prescaler is loaded to 7Fh , and the TSCR register is cleared which means that timer is stopped (PSI=0) and timer interrupt disabled.

A write to the TCR register will predominate over the 8-bit counter decrement to 00h function, i.e. if a write and a TCR register decrement to 00h occur simultaneously, the write will take precedence, and the TMZ bit is not set until the 8-bit counter reaches 00h again. The values of the TCR and the PSC registers can be read accurately at any time.

TIMERS (Continued)

Figure 30. Timer Status Control Registers



TMZ. Low-to-high transition indicates that the timer count register has decrement to zero. This bit must be cleared by user software before to start with a new count.

ETI. This bit, when set, enables the timer interrupt (vector #3 for Timer 1, vector #4 for Timer 2) request. If ETI=0 the timer interrupt is disabled. If ETI= 1 and TMZ= 1 an interrupt request is generated.

D5. This is the timers enable bit D5. It must be cleared to 0 together with a set to 1 of bit D4 to enable both Timer 1 and Timer 2 functions. It is not implemented on TSCR2 register.

D4. This is the timers enable bit D4. This bit must be set to 1 together with a clear to 0 of bit D5 to enable both Timer 1 and Timer 2 functions. It is not implemented on TSCR2 register.

D5	D4	Timers
0	0	Disabled
0	1	Enabled
1	X	Reserved

PSI. Used to initialize the prescaler and inhibit its counting while PSI = 0 the prescaler is set to 7Fh and the counter is inhibited. When PSI = 1 the prescaler is enabled to count downwards. As long as PSI= 0 both counter and prescaler are not running.

PS2-PS0. These bits select the division ratio of the prescaler register (see Table 9)

The TSCR1 and TSCR2 registers are cleared on reset. The correct D4-D5 combination must be written in TSCR1 by user's software to enable the operation of Timer 1 and Timer 2.

Table 9. Prescaler Division Factors

PS2	PS1	PS0	Divided By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Figure 31. Timer Counter Registers

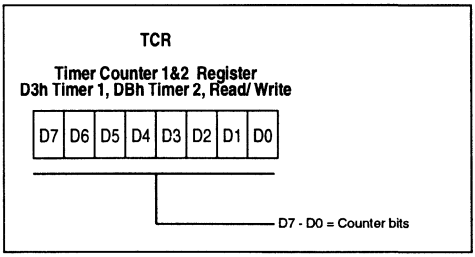
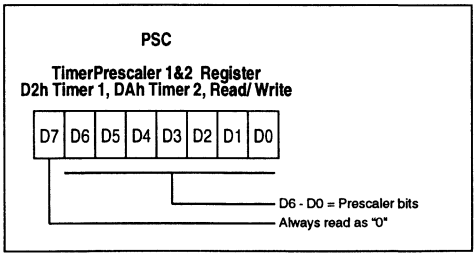


Figure 32. Timer Counter Registers



HARDWARE ACTIVATED DIGITAL WATCHDOG FUNCTION

The hardware activated digital watchdog function consists of a down counter that is automatically initialized after reset so that this function does not need to be activated by the user program. As the watchdog function is always activated this down counter can not be used as a timer. The watchdog is using one data space register (HWDR location D8h). The watchdog register is set to FEh on reset and immediately starts to count down, requiring no software start. Similarly the hardware activated watchdog can not be stopped or delayed by software.

The watchdog time can be programmed using the 6 MSbits in the watchdog register, this gives the possibility to generate a reset in a time between 3072 to 196608 oscillator cycles in 64 possible steps (With a clock frequency of 8MHz this means from 384µs to 24.576ms). The reset is prevented if the register is reloaded with the desired value before bits 2-7 decrement from all zeros to all ones.

The presence of the hardware watchdog deactivates the STOP instruction and a WAIT instruction is automatically executed instead of a STOP. Bit 1 of the watchdog register (set to one at reset) can be used to generate a software reset if cleared to zero).

Figure 34. Hardware Activated Watchdog Working Principle

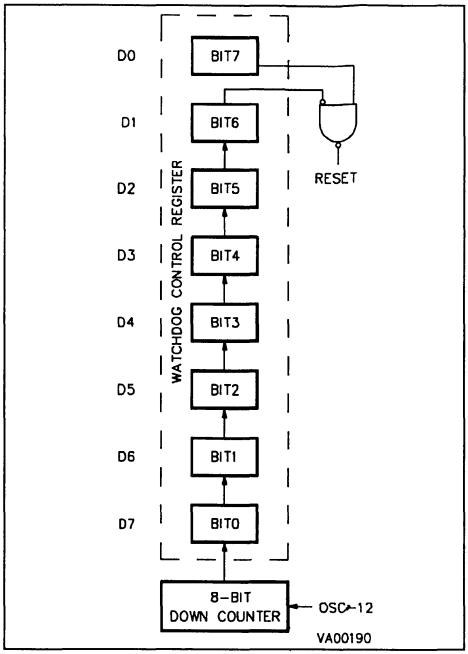
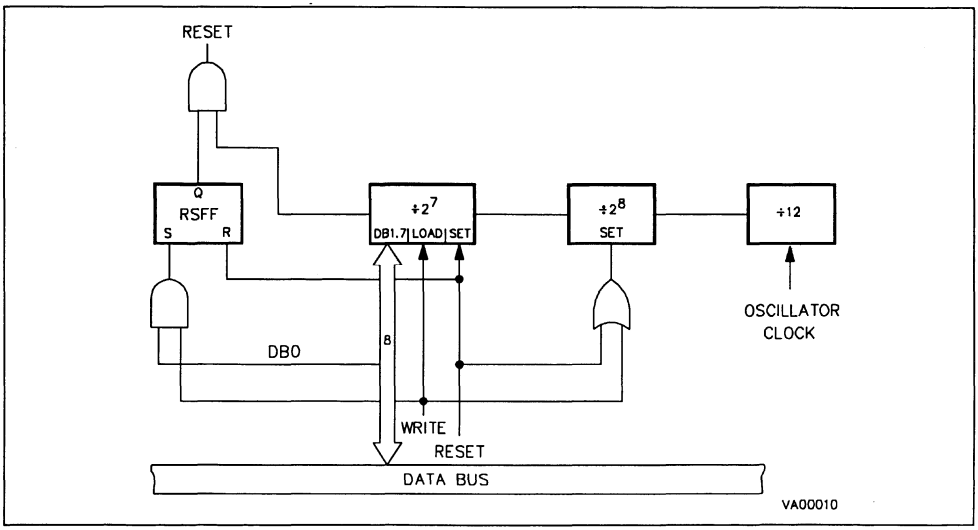
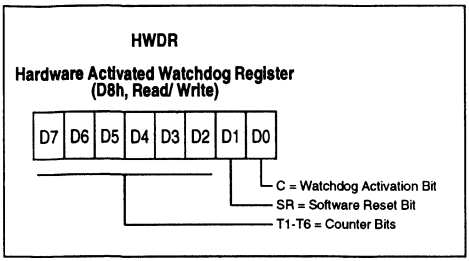


Figure 33. Hardware Activated Watchdog Block Diagram



HARDWARE ACTIVATED DIGITAL WATCHDOG FUNCTION (Continued)

Figure 35. Watchdog Register



T1-T6. These are the watchdog counter bits. It should be noted that D7 (T1) is the LSB of the counter and D2 (T6) is the MSB of the counter, these bits are in the opposite order to normal.

SR. This bit is set to one during the reset phase and will generate a software reset if cleared to zero.

C. This is the watchdog activation bit that is hardware set to one; the user can not change the value of this bit. The watchdog function is always activated independently of changes of value of this bit.

The register reset value is FEh (Bit 1-7 set to one, Bit 0 cleared).

SERIAL PERIPHERAL INTERFACE

The ST631xx Serial Peripheral Interface (SPI) has been designed to be cost effective and flexible in interfacing the various peripherals in TV applications.

It maintains the software flexibility but adds hardware configurations suitable to drive devices which require a fast exchange of data. The three pins dedicated for serial data transfer (single master only) can operate in the following ways:

- as standard I/O lines (software configuration)
- as S-BUS or as I^2CBUS (two pins)
- as standard (shift register) SPI

When using the hardware SPI, a fixed clock rate of 62.5kHz is provided.

It has to be noted that the first bit that is output on the data line by the 8-bit shift register is the MSB.

SPI Data/Control Registers

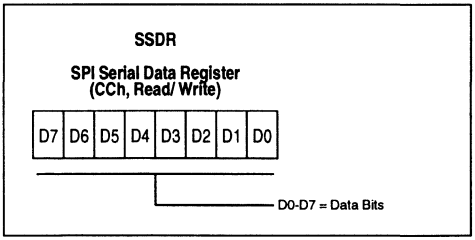
For I/O details on SCL (Serial Clock), SDA (Serial Data) and SEN (Serial Enable) please refer to I/O Ports description with reference to the following registers:

Port B data register, Address C1h (Read/Write).

- BIT D5 "SCL"
- BIT D6 "SDA"
- BIT D7 "SEN"

Port B data direction register, Address C5h (Read/Write).

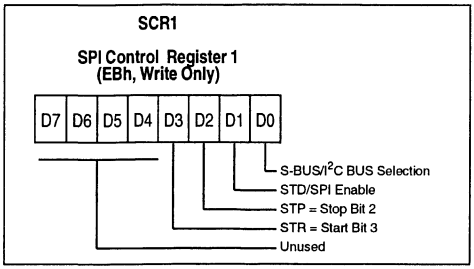
Figure 36. SPI Serial Data Register



D7-D0. These are the SPI data bits. They can be neither read nor written when SPI is operating (BUSY bit set). They are undefined after reset.

SERIAL PERIPHERAL INTERFACE (Continued)

Figure 37. SPI Control Register 1



D7-D4. These bits are not used.

STR. This is Start bit for I^2CBUS/S-BUS. This bit is meaningless when STD/SPI enable bit is cleared to zero. If this bit is set to one STD/SPI bit is also set to "1" and SPI Start generation, before beginning of transmission, is enabled. Set to zero after reset.

STP. This is Stop bit for I^2CBUS/S-BUS. This bit is meaningless when STD/SPI enable bit is cleared to zero. If this bit is set to one STD/SPI bit is also set to "1" and SPI Stop condition generation is enabled. STP bit must be reset when standard protocol is used (this is also the default reset conditions). Set to zero after reset.

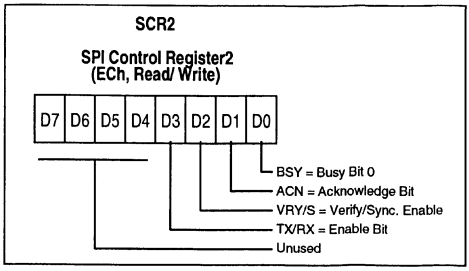
STD, SPI Enable. This bit, in conjunction with S-BUS/I^2CBUS bit, allows the SPI disable and will select between I^2CBUS/S-BUS and Standard shift register protocols. If this bit is set to one, it selects both I^2CBUS and S-BUS protocols; final selection between them is made by S-BUS/I^2CBUS bit. If this bit is cleared to zero when S-BUS/I^2CBUS is set to "1" the Standard shift register protocol is selected. If this bit is cleared to "0" when S-BUS/I^2CBUS is cleared to 0 the SPI is disabled. Set to zero after reset.

S-BUS/I^2CBUS Selection. This bit, in conjunction with STD/SPI bit, allows the SPI disable and will select between I^2CBUS and S-BUS protocols. If this bit is cleared to "0" when STD bit is also "0", the SPI interface is disabled. If this bit is cleared to zero when STD bit is set to "1", the I^2CBUS protocol will be selected. If this bit is set to "1" when STD bit is set to "1", the S-BUS protocol will be selected. Cleared to zero after reset.

Table 10. SPI Modes Selection

D1 STD/SP	D0 S-BUS/I^2C BUS	SPI Function
0	0	Disabled
0	1	STD Shift Reg.
1	0	I^2C BUS
1	1	S-BUS

Figure 38. SPI Control Register 2



D7-D4. These bits are not used.

TX/RX. *Write Only.* When this bit is set, current byte operation is a transmission. When it is reset, current operation is a reception. Set to zero after reset.

VRY/S. *Read Only/Write Only.* This bit has two different functions in relation to read or write operation. *Reading Operation:* when STD and/or TRX bits is cleared to 0, this bit is meaningless. When bits STD and TX are set to 1, this bit is set each time BSY bit is set. This bit is reset during byte operation if real data on SDA line are different from the output from the shift register. Set to zero after reset. *Writing Operation* : it enables (if set to one) or disables (if cleared to zero) the interrupt coming from VSYNC pin. Undefined after reset. Refer to OSD description for additional information.

ACN. *Read Only.* If STD bit (D1 of SCR1 register) is cleared to zero this bit is meaningless. When STD is set to one, this bit is set to one if no Acknowledge has been received. In this case it is automatically reset when BSY is set again. Set to zero after reset.

BSY. *Read/Set Only.* This is the busy bit. When a one is loaded into this bit the SPI interface start the transmission of the data byte loaded into SSDR data register or receiving and building the receive data into the SSDR data register. This is done in accordance with the protocol, direction and start/stop condition(s). This bit is automatically cleared at the end of the current byte operation. Cleared to zero after reset.

Note :
The SPI shift register is also the data transmission register and the data received register; this feature is made possible by using the serial structure of the ST631xx and thus reducing size and complexity.

SERIAL PERIPHERAL INTERFACE (Continued)

During transmission or reception of data, all access to serial data register is therefore disabled. The reception or transmission of data is started by setting the BUSY bit to "1"; this will be automatically reset at the end of the operation. After reset, the busy bit is cleared to "0", and the hardware SPI disabled by clearing bit 0 and bit 1 of SPI control register 1 to "0". The outputs from the hardware SPI are "ANDed" to the standard I/O software controlled outputs. If the hardware SPI is in operation the Port C pins related to the SPI should be configured as outputs using the Data Direction Register and should be set high. When the SPI is configured as the S-BUS, the three pins PC0, PC1 and PC3 become the pins SCL, SDA and SEN respectively. When configured as the I^2CBUS the pins PC0 and PC1 are configured as the pins SCL and SDA; PC3 is not driven and can be used as a general purpose I/O pin. In the case of the STD SPI the pins PC0 and PC1 become the signals CLOCK and DATA, PC3 is not driven and can be used as general purpose I/O pin. The VERIFY bit is available when the SPI is configured as either S-BUS or I^2CBUS. At the start of a byte transmission, the verify bit is set to one. If at any time during the transmission of the following eight bits, the data on the SDA line does not match the data forced by the SPI (while SCL is high), then the VERIFY bit is reset. The verify is available only during transmission for the S-BUS and I^2CBUS; for other protocol it is not defined. The SDA and SCL signal entering the SPI are buffered in order to remove any minor glitches. When STD bit is set to one (S-BUS or I^2CBUS selected), and TRX bit is reset (receiving data), and STOP bit is set (last byte of current communication), the SPI interface does not generate the Acknowledge, according to S-BUS/I^2CBUS specifications. PCO-SCL, PC1-SDA and PC3-SEN lines are standard drive I/O port pins with open-drain output configuration (maximum voltage that can be applied to these pins is V_{DD}+ 0.3V).

S-BUS/I^2CBUS Protocol Information

The S-BUS is a three-wire bidirectional data-bus with functional features similar to the I^2CBUS. In fact the S-BUS includes decoding of Start/Stop conditions and the arbitration procedure in case of multimaster system configuration (the ST631xx SPI allows a single-master only operation). The SDA line, in the I^2CBUS represents the AND combination of SDA and SEN lines in the S-BUS. If the SDA and the SEN lines are short-circuit connected, they appear as the SDA line of the I^2CBUS. The Start/Stop conditions are detected (by the external peripherals suited to work with S-BUS/I^2CBUS) in the following way:

- On S-BUS by a transition of the SEN line (1 to 0 Start, 0 to 1 Stop) while the SCL line is at high level.
- On I^2CBUS by a transition of the SDA line (10 Start, 01Stop) while the SCL line is at high level.

Start and Stop condition are always generated by the master (ST631xx SPI can only work as single master). The bus is busy after the start condition and can be considered again free only when a certain time delay is left after the stop condition. In the S-BUS configuration the SDA line is only allowed to change during the time SCL line is low. After the start information the SEN line returns to high level and remains unchanged for all the data transmission time. When the transmission is completed the SDA line is set to high level and, at the same time, the SEN line returns to the low level in order to supply the stop information with a low to high transition, while the SCL line is at high level. On the S-BUS, as on the I^2CBUS, each eight bit information (byte) is followed by one acknowledged bit which is a high level put on the SDA line by the transmitter. A peripheral that acknowledges has to pull down the SDA line during the acknowledge clock pulse. An addressed receiver has to generate an acknowledge after the reception of each byte; otherwise the SDA line remains at the high level during the ninth clock pulse time. In this case the master transmitter can generate the Stop condition, via the SEN (or SDA in I^2CBUS) line, in order to abort the transfer.

SERIAL PERIPHERAL INTERFACE (Continued)

Start/Stop Acknowledge. The timing specs of the S-BUS protocol require that data on the SDA (only on this line for I^2CBUS) and SEN lines be stable during the "high" time of SCL. Two exceptions to this rule are foreseen and they are used to signal the start and stop condition of data transfer.

- On S-BUS by a transition of the SEN line (10 Start, 01 Stop) while the SCL line is at high level.
- On I^2CBUS by a transition of the SDA line (10 Start, 01 Stop) while the SCL line is at high level.

Data are transmitted in 8-bit groups; after each group, a ninth bit is interposed, with the purpose of acknowledging the transmitting sequence (the transmit device place a "1" on the bus, the acknowledging receiver a "0").

Interface Protocol. This paragraph deals with the description of data protocol structure. The interface protocol includes:

- A start condition

- A "slave chip address" byte, transmitted by the master, containing two different information:

a. the code identifying the device the master wants to address (this information is present in the first seven bits)

b. the direction of transmission on the bus (this information is given in the 8th bit of the byte); "0" means "Write", that is from the master to the slave, while "1" means "Read". The addressed slave must always acknowledge.

The sequence from, now on, is different according to the value of R/$\overline{W}$ bit.

1. R/$\overline{W}$ = "0" ($\overline{\text{Write}}$)

In all the following bytes the master acts as transmitter; the sequence follows with:

a. an optional data byte to address (if needed) the slave location to be written (it can be a word address in a memory or a register address, etc.).

b. a "data" byte which will be written at the address given in the previous byte.

c. further data bytes.

d. a STOP condition

A data transfer is always terminated by a stop condition generated from the master. The ST631xx peripheral must finish with a stop condition before another start is given. Figure 44 shows an example of write operation.

2. R/$\overline{W}$ = "1" (Read)

In this case the slave acts as transmitter and, therefore, the transmission direction is changed. In read mode two different conditions can be considered:

a. The master reads slave immediately after first byte. In this case after the slave address sent from the master with read condition enabled the master transmitter becomes master receiver and the slave receiver becomes slave transmitter.

b. The master reads a specified register or location of the slave. In this case the first sent byte will contain the slave address with write condition enabled, then the second byte will specify the address of the register to be read. At this moment a new start is given together with the slave address in read mode and the procedure will proceed as described in previous point "a".

SERIAL PERIPHERAL INTERFACE (Continued)

Figure 39.Master Transmit to Slave Receiver (Write Mode)

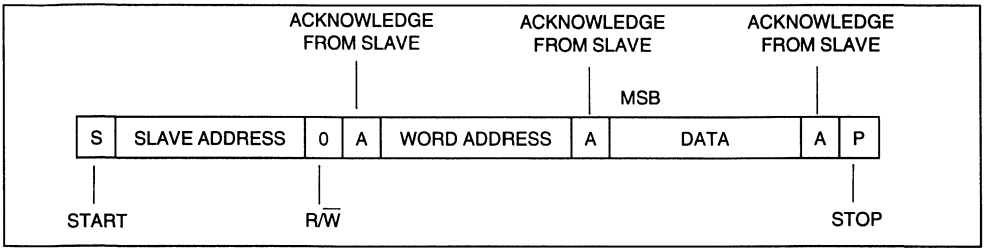


Figure 40.Master Reads Slave Immediately After First Byte (read Mode)

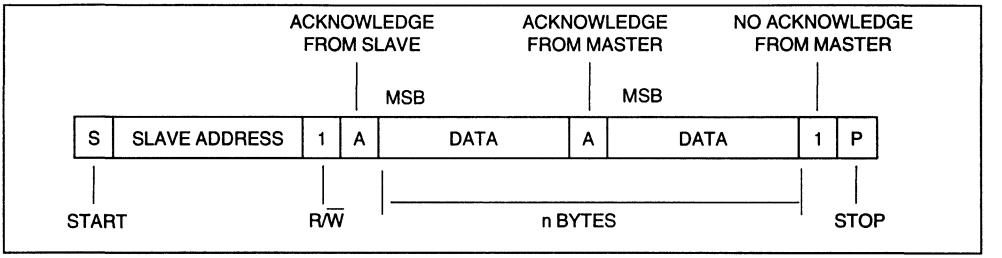
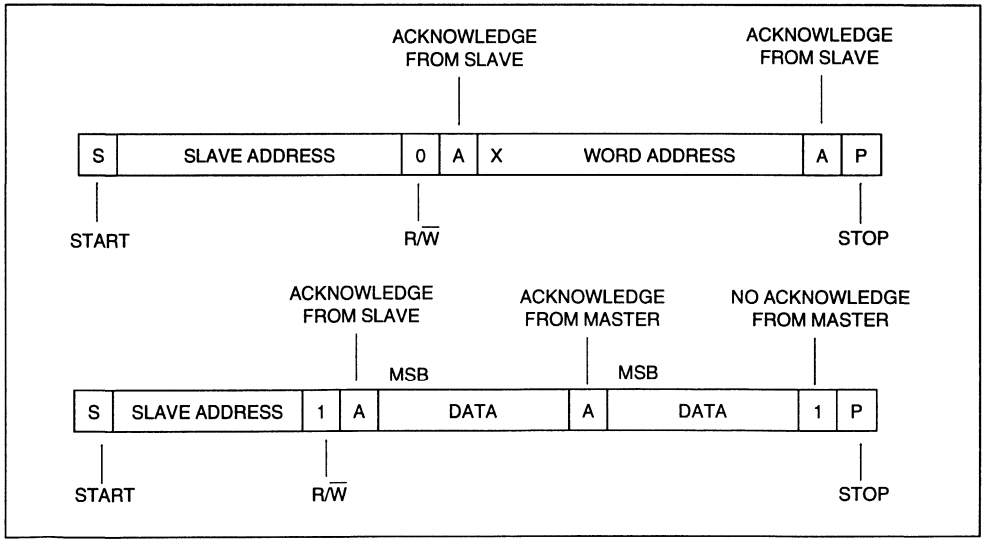


Figure 41.Master Reads After Setting Slave Register Address (Write Address, Read Data)

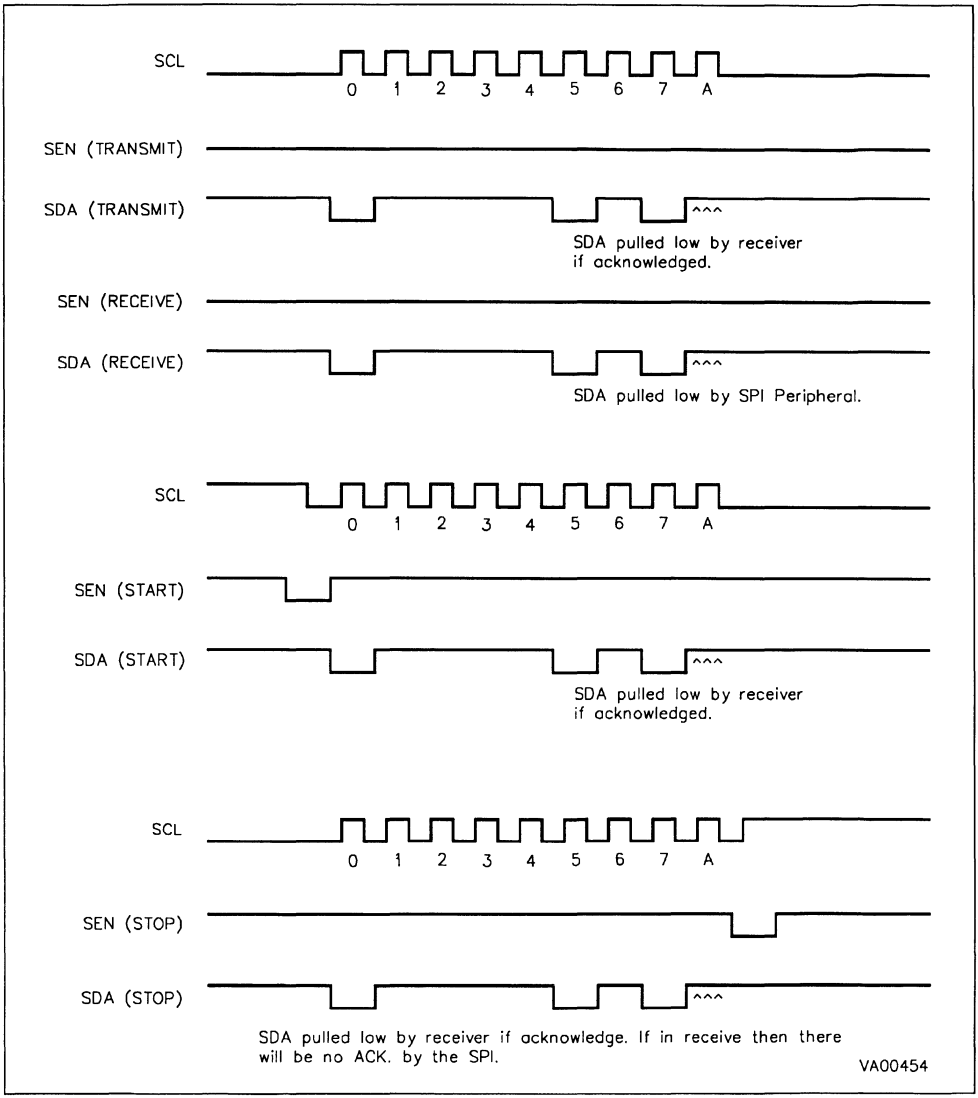


SERIAL PERIPHERAL INTERFACE (Continued)

S-BUS/I²CBUS Timing Diagrams

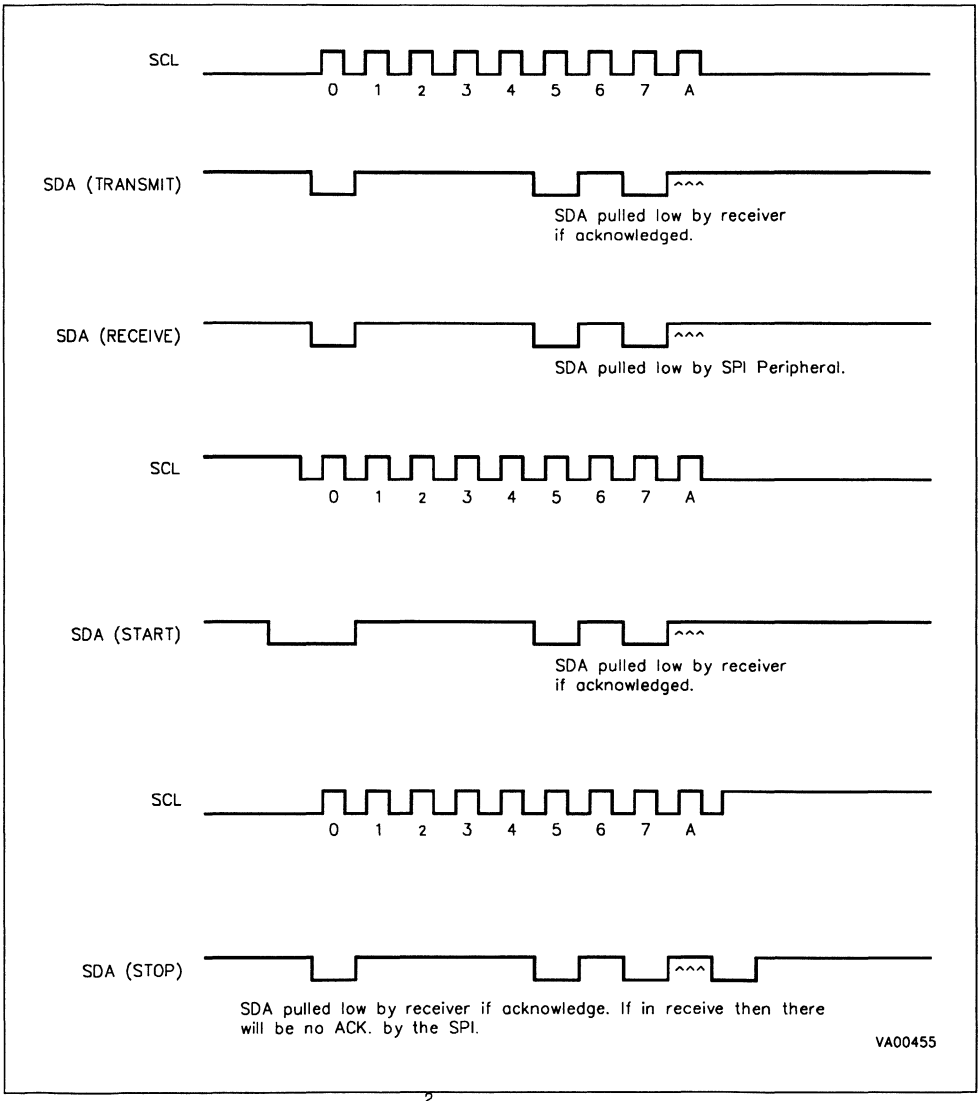
The clock of the S-BUS/I²CBUS of the ST631xx SPI (single master only) has a fixed bus clock frequency of 62.5kHz. All the devices connected to the bus must be able to follow transfers with frequencies up to 62.5kHz, either by being able to transmit or receive at that speed or by applying the clock synchronization procedure which will force the master into a wait state and stretch low periods.

Figure 42. S-BUS Timing Diagram



SERIAL PERIPHERAL INTERFACE (Continued)

Figure 43. I^2C BUS Timing Diagram



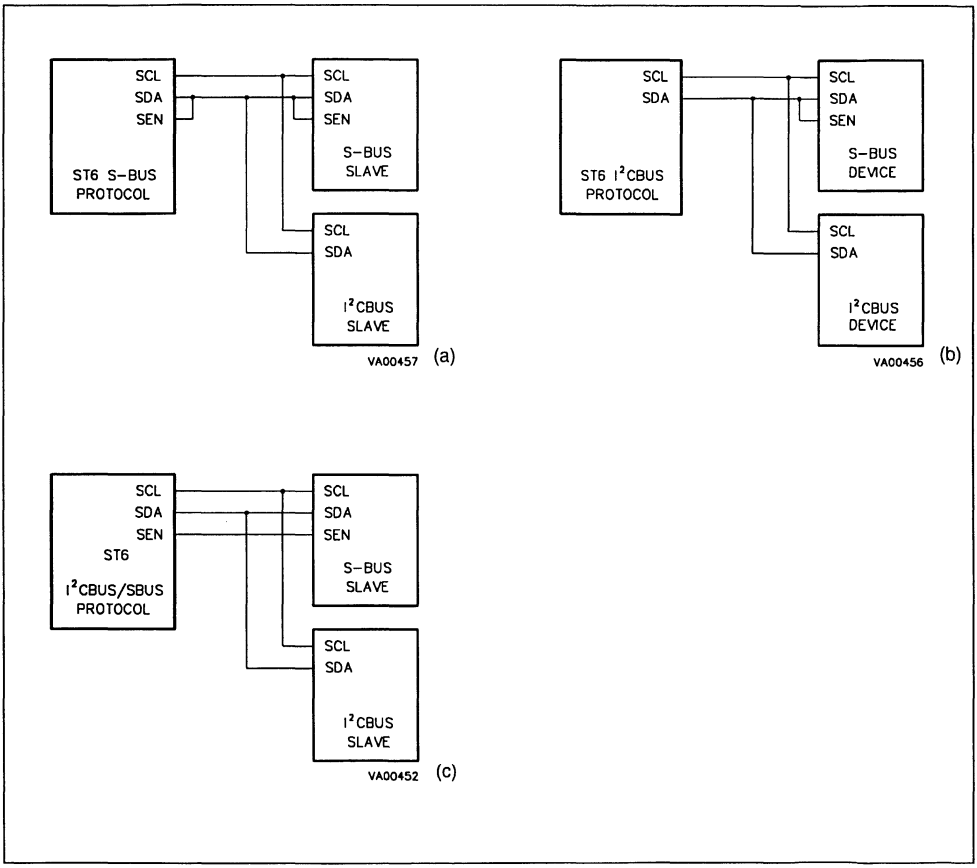
Note: The third pin, SEN, should be high; it is not used in the I^2CBUS. Logically SDA is the AND of the S-BUS SDA and SEN.

SERIAL PERIPHERAL INTERFACE (Continued)

Compatibility S-BUS/I²CBUS

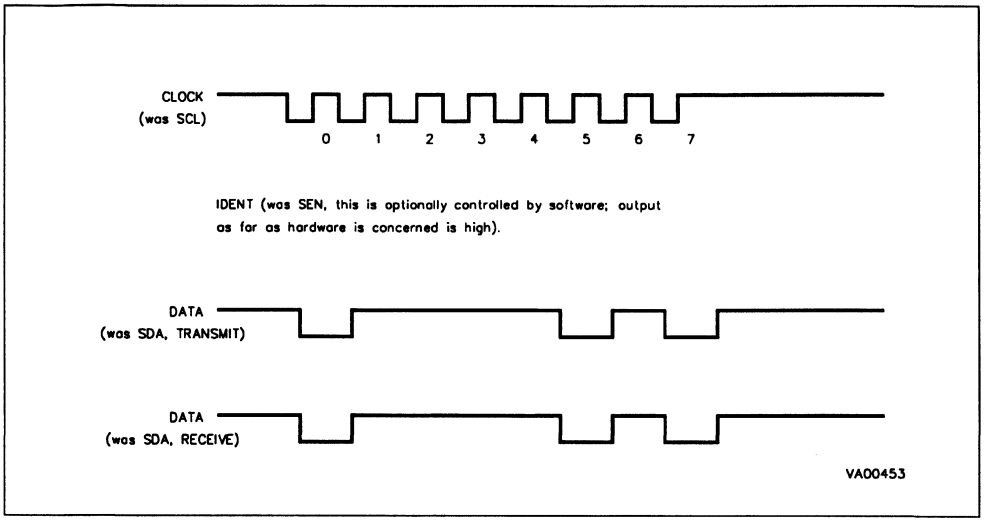
Using the S-BUS protocol it is possible to implement mixed system including S-BUS/I²CBUS bus peripherals. In order to have the compatibility with the I²CBUS peripherals, the devices including the S-BUS interface must have their SDA and SEN pins connected together as shown in the following Figure 44 (a and b). It is also possible to use mixed S-BUS/I²CBUS protocols as showed in Figure 48 (c). S-BUS peripherals will only react to S-BUS protocol signals, while I²CBUS peripherals will only react to I²CBUS signals. Multimaster configuration is not possible with the ST631xx SPI (single master only).

Figure 44.S-BUS/I²C BUS Mixed Configurations



SERIAL PERIPHERAL INTERFACE (Continued)

Figure 45.STD Bus (Hardware Bus Disabled) Timing Diagram



STD SPI Protocol (Shift Register)

This protocol is similar to the I^2CBUS with the exception that there is no acknowledge pulse and there are no stop or start bits. The clock cannot be slowed down by the external peripherals.

The I/O ports associated with the SPI should be programmed as outputs with data high in order not to inhibit the functionality of the hardware SPI.

SPI APPLICATION NOTES

Stop Clock Slowdown: In the ST631xx family of devices when operating in the I^2C or SBUS modes, there is no internal clock slowdown for the final STOP clock. Slowdown means that if an external peripheral requires extra time it will hold the ST631xx SCL clock low. To be fully I^2C and SBUS compatible in this respect, the SW should check that the SCL line is indeed high before proceeding with the START of another I^2C or SBUS transmission. In all other cases the SCL clock slowdown feature is operational.

SPI Standard Bus Protocol: The standard bus protocol is selected by loading the SPI Control Register 1 (SCR1 Add. EBh). Bit 0 named I^2C must be set at one and bit 1 named STD mut be reset. When the standard bus protocol is selected bit 2 of the SCR1 is meaningless.

This bit named STOP bit is used only in I^2CBUS or SBUS. However take care thet THE STOP BIT MUST BE RESET WHEN THE STANDARD PROTOCOL IS USED. This bit is set to ZERO after RESET.

14-BIT VOLTAGE SYNTHESIS TUNING PERIPHERAL

The ST631xx on-chip voltage synthesis tuning peripheral has been integrated to allow the generation of tuning reference voltage in low/mid end TV set applications. The peripheral is composed of a 14-bit counter that allows the conversion of the digital content in a tuning voltage, available at the VS output pin, by using Pulse Width Modification (PWM), and Bit Rate Multiplier (BRM) techniques. The 14-bit counter gives 16384 steps which allows a resolution of approximately 2mV over a tuning voltage of 32V; this corresponds to a tuning resolution of about 40kHz per step in the UHF band (the actual value will depend on the characteristics of the tuner).

The tuning word consists of a 14-bit word contained in the registers VSDATA1 (location 0EDh) and VSDATA2 (location 0EEh). Coarse tuning (PWM) is performed using the seven MSBits, while fine tuning (BRM) is performed using the data in the seven LSBits. With all zeros loaded the output is zero; as the tuning voltage increases from all zeros, the number of pulses in one period increas to 128 with all pulses being the same width. For values larger than 128, the PWM takes over and the number of pulses in one period remains constant at 128, but the width changes. At the other end of the scale, when almost all ones are loaded, the pulses will start to link together and the number of pulses will decrease. When all ones are loaded, the output will be almost 100% high but will have a low pulse (1/16384 of the high pulse).

Output Details

Inside the on-chip Voltage Synthesis are included the register latches, a reference counter, PWM and BRM control circuitry. In the ST631xx the clock for the 14-bit reference counter is 2MHz derived from the 8MHz system clock. From the circuit point of view, the seven most significant bits control the coarse tuning, while the seven least significant bits control the fine tuning. From the application and software point of view, the 14 bits can be considered as one binary number.

As already mentioned the coarse tuning consists of a PWM signal with 128 steps; we can consider the fine tuning to cover 128 coarse tuning cycles. The addition of pulses is described in the following Table.

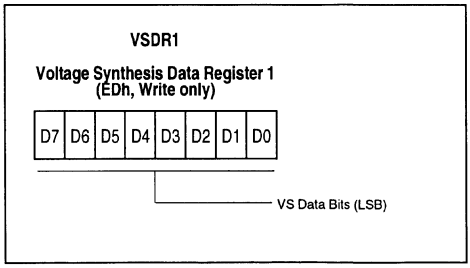
Table 11. Fine Tuning Pulse Addition

Fine Tuning (7 LSB)	N° of Pulses added at the following cycles (0...127)
0000001	64
0000010	32, 96
0000100	16, 48, 80, 112
0001000	8, 24, ...104, 120
0010000	4, 12, ...116, 124
0100000	2, 6, ...122, 126
1000000	1, 3, ...125, 127

The VS output pin has a standard drive push-pull output configuration.

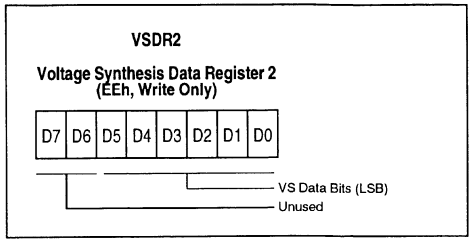
VS Tuning Cell Registers

Figure 46. Voltage Synthesis Data Register 1



D7-D0. These are the 8 least significant VS data bits. Bit 0 is the LSB. This register is undefined on reset.

Figure 47. Voltage Synthesis Data Register 2



D7-D6. These bits are not used.

D5-D0. These are the 6 most significant VS data bits. Bit 5 is the MSB. This register is undefined on reset.

6-BIT PWM D/A CONVERTER AND 62.5 kHz OUTPUT FUNCTION

The D/A macrocell contains four PWM D/A outputs (31.25kHz repetition, DA0-DA3) with six bit resolution plus a 62.5kHz open-drain output pin (OUT1) specially suited for multistandard chroma processors driving. Both the D/A and OUT1 functions can be disabled by software allowing the DA0-DA3 and OUT1 pins to be used as general purpose open-drain output pins able to withstand signals with up to 12V amplitude.

6-Bit D/A Converters

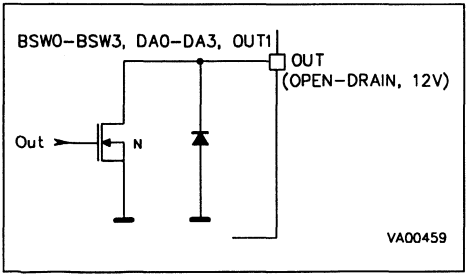
Each D/A converter of ST631xx is composed by the following main blocks:

- pre-divider
- 6-bit counter
- data latches and compare circuits

The pre-divider uses the clock input frequency (8MHz) and its output clocks the 6-bit free-running counter. The data latched in the four registers (E0h, E1h, E2h and E3h) control the four D/A outputs (DA0,1,2 and 3). When all zeros are loaded the relevant output is an high logic level; all 1's correspond to a pulse with a 1/64 duty cycle and almost 100% zero level. A 7th bit (bit D6) is used to enable the relevant D/A output; when zero, the D/A is no longer enabled and it forces the output to zero. If the other six bits are all zero then the output is controlled only by the enable bit.

The repetition frequency is 32.5kHz and is related to the 8MHz clock frequency. All D/A outputs are open-drain with standard current drive capability and able to withstand up to 12V.

Figure 48. 6-bit PWM D/A & 62.5kHz Output Configuration



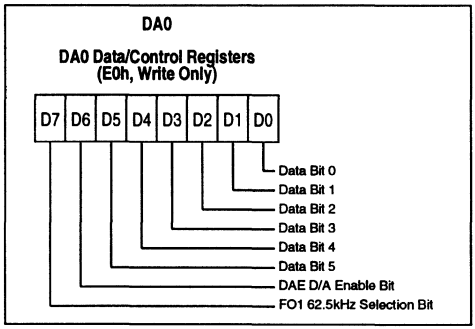
62.5 kHz Output

This pin provides a 62.5 kHz signal with a 50% duty cycle; the output is enabled by a dedicated enable bit (E0h register bit 7). When the 62.5kHz frequency is disabled then the output is controlled by the OUT1 bit and the line can be used as general purpose open-drain output (E1h bit 7). The OUT1 output is open-drain with standard current drive capability and able to withstand signals with up to 12V amplitude.

D/A and OUT1 Data/Control Registers

This paragraph deals with the description of D/A and OUT1 data/control registers. Some bits of DA2 and DA3 data/control registers are used for external interrupt enable and A/D reference voltage shift, please refer to A/D and IR descriptions for additional information.

Figure 49. DA0 Data/Enable Register



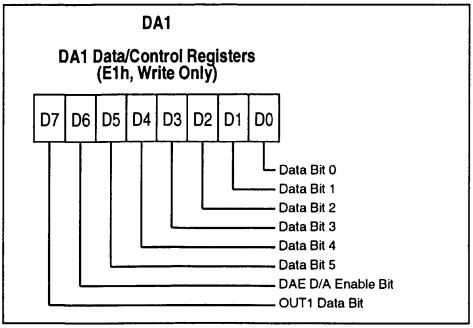
DA0-DA5. These are the 6 bits of the PWM digital to analog converter . Undefined after reset.

DAE. This is the D/A 0 enable bit. If zero, the output of the D/A is forced to zero; if one, the output of the D/A depends on bits DA0..DA5. Undefined after reset.

FO1. This is the 62.5kHz frequency output/ OUT1 selection bit. If one, the OUT1 pin will give a 62.5kHz frequency; if zero the OUT1 pin can be used as general purpose open-drain output and the value present on the pin depends on the value of OUT1 bit programmed in the DA1 data/control register. Undefined after reset.

6-BIT PWM D/A CONVERTERS AND 62.5 kHz OUTPUT FUNCTION (Continued)

Figure 50. DA1 Data/Enable Register

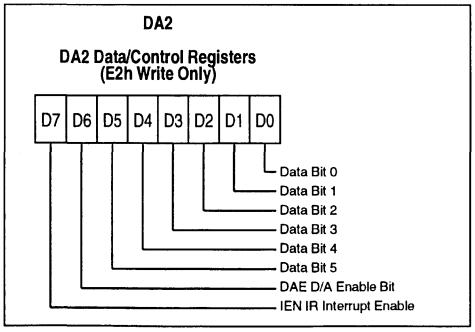


DA0-DA5. These are the 6 bits of the PWM digital to analog converter. Undefined after reset.

DAE. This is the D/A 1 enable bit. If zero, the output of the D/A is forced to zero; if one, the output of the D/A depends on bits DA0..DA5. Undefined after reset.

OUT1. This is the OUT1 data bit. The content of this bit is output on the OUT1 pin when the 62.5kHz frequency function is disabled (FO1 bit in DA0 register is cleared to zero). Undefined after reset.

Figure 51. DA2 Data/Enable Register

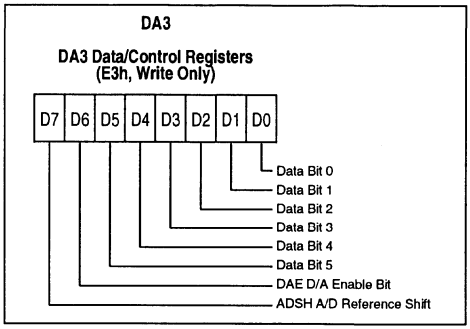


DA0-DA5. These are the 6 bits of the PWM digital to analog converter bits. Undefined after reset.

DAE. This is the D/A 2 enable bit. If zero, the output of the D/A is forced to zero; if one, the output of the D/A depends on bits DA0..DA5. Undefined after reset.

IEN. This is the external interrupt enable. If set to one, the interrupt coming from the external interrupt pin is enabled, if this bit is cleared the interrupt is disabled. Undefined after reset. This interrupt is associated to the NMI interrupt vector. Refer to IR and interrupt descriptions for additional information.

Figure 52. DA3 Data/Enable Register



DA0-DA5. These are the 6 bits of the PWM digital to analog converter. Undefined after reset.

DAE. This is the D/A 3 enable bit. If zero, the output of the D/A is forced to zero; if one, the output of the D/A depends on bits DA0..DA5. Undefined after reset.

ADSH. This is the analog to digital converter reference voltage shift bit. If set to one, the AFC block has reference voltages on 1V border. If set to zero, on 0.5V border. Undefined after reset. Refer to AFC for additional information.

AFC A/D INPUT, KEYBOARD INPUTS AND BANDSWITH OUTPUTS

The AFC macrocell contains an A/D comparator with five levels at intervals of 1V from 1V to 5V. The levels can all be lowered by 0.5V to effectively double the resolution. This A/D can be used to perform the AFC function. In addition this cell offers also a keyboard input register of three bits used to perform a keyboard scan and 4 open-drain outputs (able to withstand signals up to 12V) that can be used to perform band switch function.

Figure 53. AFC, KBY Inputs Configuration Diagrams

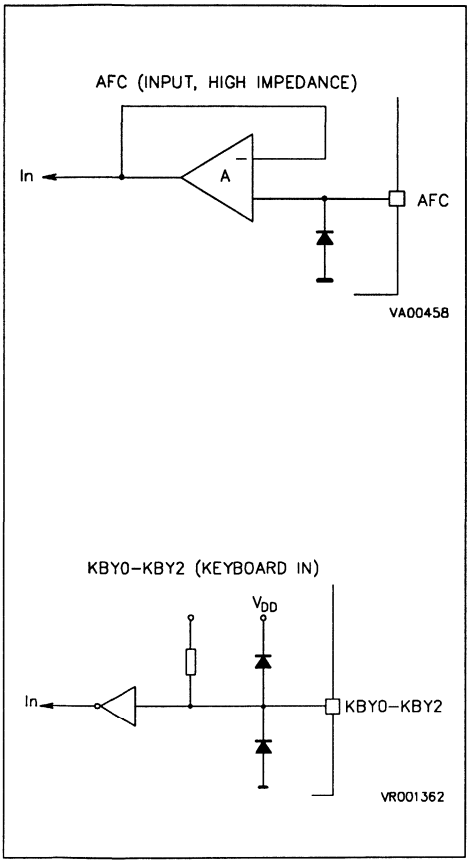
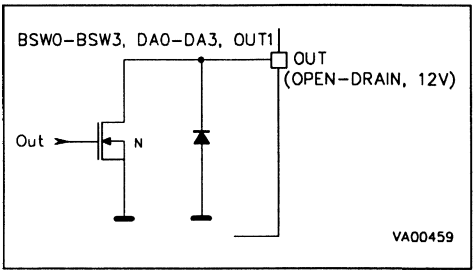


Figure 54. BSW, DA, OUT1 Output Configuration Diagram



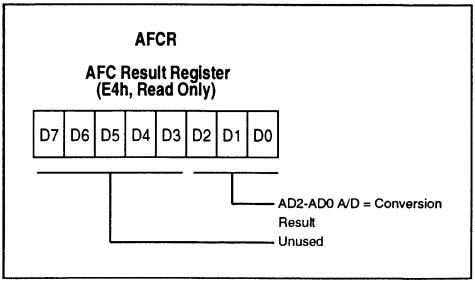
A/D Comparator

The A/D used to perform the AFC function (when high threshold is selected) has the following voltage levels: 1,2,3,4 and 5V. Bits 0-2 of AFC result register (E4h address) will provide the result in binary form (less than 1V is 000, greater than 5V is 101).

If the application requires a greater resolution, the sensitivity can be doubled by clearing to zero bit 7 of DA3 Data/Control register, address E3h (refer to D/A description for additional information). In this case all levels are shifted lower by 0.5V. If the two results are now added within a software routine then the A/D S-curve can be located within a resolution of 0.5V. The A/D input has high impedance able to withstand up to 13V signals (input level tolerances ± 200mv absolute and ± 100mv relative to 5V).

AFC, Keyboard Inputs and Bandswitch Outputs Data/Control Registers

Figure 55. AFC Result Register

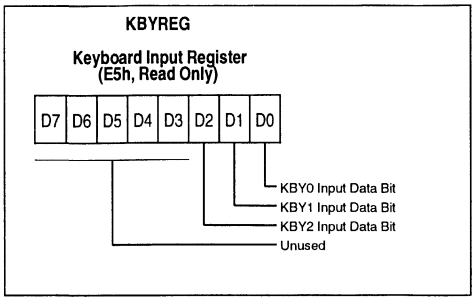


D7-D3. These bits are not used.

AD0-AD2. These bits store the real time conversion of the value present on the AFC input pin. No reset value.

AFC A/D INPUT, KEYBOARD INPUTS AND BANDSWITH OUTPUTS (Continued)

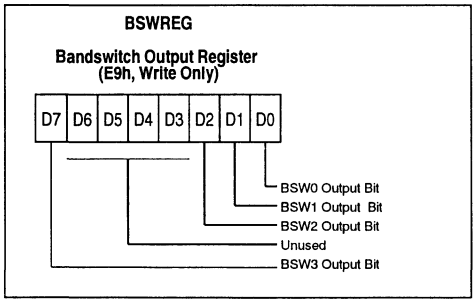
Figure 56. Keyboard Input Register



D7-D3. These bits are not used.

KBY0-KBY2. These bits store the logic level present at KBY0, KBY1 and KBY2 input pins. No reset value. This input pins have CMOS levels with on-chip pull-up resistor (100kΩ typical).

Figure 57. Bandswitch Output Register



D6-D3. These bits are not used.

BSW0-BSW2,BSW3. The writing into these bits will cause the corresponding BSW open-drain output line to switch to the programmed level. Undefined after reset.

INFRARED INPUT (IRIN)

The IRIN pin is directly connected to the NMI interrupt and acts as external interrupt pin (refer to interrupt description for additional information).

The enable/disable of this interrupt can be managed with the write only IEN bit available in the DA2 Data/Control Register (Address E2h, bit D7). When this bit is set to one the interrupt is enabled otherwise it is disabled.

The IRIN pin is RISING EDGE sensitive.

Application Note

When the IR interrupt is enabled, then a rising edge on the IR pin will generate an interrupt; if the IR interrupt is disabled, no IR interrupts can occur. Care should be taken because if the IR pin is high when the IR interrupt is enabled, an interrupt will also be generated; the following method to eliminate noise can also be used if the SW engineer wishes to enable/disable the IR interrupt.

If a Low-cost infra-red receiver is used, the customer may wish to test the IR signal by software after an interrupt in order to verify that there is a good pulse and not just noise. The IRIN pin cannot be read, so in this case it should be connected in parallel with another pin so the signal can be read. Furthermore the IRIN pin is sensitive to a rising edge interrupt; this means that the input to the pin should be low in the presence of no infra-red signal, but since most infra-red receiver modules give a high signal, the signal will need to be inverted with a transistor.

ON-SCREEN DISPLAY (OSD)

The ST631xx OSD macrocell is a CMOS LSI character generator which enable display of characters and symbols on the TV screen. The character rounding function enhances the readability of the characters. The ST631xx OSD receives horizontal and vertical synchronization signal and outputs screen information via R, G, B and blanking pins. The main characteristics of the macrocell are listed below:

- Number of display characters: 5 lines by 15 columns.
- Number of character types: 128 characters in two banks of 64 characters. **Only one bank per screen can be used.**
- Character size: Four character heights (18h, 36h 54h, 72h), two heights are available per screen, programmable by line.
- Character format: 6x9 dots with character rounding function.
- Character colour: Eight colours available programmable by word.
- Display position: 64 horizontal positions by $2/f_{osc}$ and 63 vertical positions by 4 h
- Word spacing: 64 positions programmable from $2/f_{osc}$ to $128/f_{osc}$.
- Line spacing: 63 positions programmable from 4 to 252 h.
- Background: No background, square background or fringe background programmable by word.
- Background colour: Two of eight colours available programmable by word.
- Display output: Three character data output terminals (R,G,B) and a blank output terminal.
- Display on/off: Display data may be programmed on or off by word or entire screen. The entire screen may be blanked.

Format Specification

The entire display can be turned on or off through the use of the global enable bit or the display may be selectively turned on or off by word. To turn off the entire display, the global enable bit (GE) should be zero. If the global enable is one, the display is controlled by the word enable bits (WE). The global enable bit is located in the global enable register and the word enable bit is located in the space character preceding the word.

Each line must begin with a format character which describes the format of that line and of the first word. This character is not displayed.

A space character defines the format of subsequent words. A space character is denoted by a one in bit 6 in the display RAM. If bit 6 of the display RAM is a zero, the other six bits define one of the 64 display characters.

The colour, background and enable can be programmed by word. This information is encoded in the space character between words or in the format character at the beginning of each line. Five bits define the colour and background of the following word, and determine whether it will be displayed or not.

Characters are stored in a 6 x 9 dot format. One dot is defined vertically as 2h (horizontal lines) and horizontally as $2/f_{osc}$ if the smallest character size is enabled. There is no space between characters or lines if the vertical space enable (VSE) and horizontal space enable (HSE) bits are both zero. This allows the use of special graphics characters.

The normal alphanumeric character set is formatted to be 5 x 7 with one empty row at the top and one at the bottom and one empty column at the right. If VSE and HSE are both zero, then the spacing between alphanumeric characters is 1 dot and the spacing between lines of alphanumeric characters is 2h.

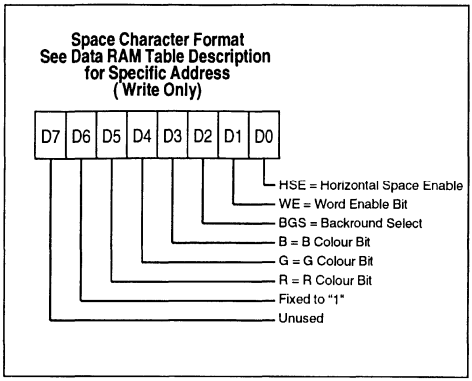
The character size is programmed by line through the use of the size bit (S) in the format character and the global size bits (GS1 and GS2). The vertical spacing enable bit (VSE) located in the format character controls the spacing between lines. If this bit is set to one, the spacing between lines is defined by the vertical spacing register, otherwise the spacing between lines is 0.

The spacing between words is controlled by the horizontal space enable bit (HSE) located in the space character. If this bit is set to one, the spacing between words is defined by the horizontal spacing register, otherwise the space character width of 6 dots is the spacing between words.

The formats for the display character, space character and format character are described hereafter.

ON-SCREEN DISPLAY (Continued)

Figure 58. Space Character Register Explanation



D7. Not used.

D6. This pin is fixed to "1".

R, G, B. *Colour.* The 3 colour control bits define the colour of the following word as shown in table below.

Space Character Register Colour Setting.

R	G	B	Colour
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

BGS. *Background Select.* The background select bit selects the desired background for the following word. There are two possible backgrounds defined by the bits in the Background Control Register.

"0" -The background on the following word is enabled by BG0 and the colour is set by R0, G0, and B0.

"1" -The background on the following word is enabled by BG1 and the colour is set by R1, G1, and B1.

WE. *Word Enable.* The word enable bit defines whether or not the following word is displayed.

"0" -The word is not displayed.

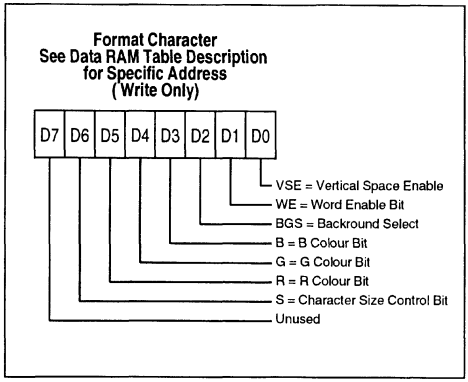
"1" -If the global enable bit is one, then the word is displayed.

HSE. *Horizontal Space Enable.* The horizontal space enable bit determines the spacing between words. The space between characters is always 0. The alphanumeric character set is implemented in a 5 x 7 format with one empty column to the right and one empty row above and below so that the space between alphanumeric characters will be one dot.

"0" -The space between words is equal to the width of the space character, which is 6 dots.

"1" -The space between words is defined by the value in the horizontal space register plus the width of the space character.

Figure 59. Format Character Register Explanation



D7. This bit is not used

S. *Character Size.* The character size bit, along with the global size bits (GS2 and GS1) located in the horizontal space register, specify the character size for each line as defined in Table 14.

R, G, B. *Colour.* The 3 colour control bits define the colour of the following word as shown in Table 13.

BGS. *Background Select.* The background select bit selects the desired background for the following word. There are two possible backgrounds defined by the bits in the Background Control Register.

"0" -The background on the following word is enabled by BG0 and the colour is set by R0, G0, and B0.

ON-SCREEN DISPLAY (Continued)

"1" -The background on the following word is enabled by BG1 and the colour is set by R1, G1, and B1.

WE. *Word Enable.* The word enable bit defines whether or not the following word is displayed.

"0" -The word is not displayed.

"1" -If the global enable bit is one, then the word is displayed.

VSE. *Vertical Space Enable.* The vertical space enable bit determines the spacing between lines.

"0" -The space between lines is equal to 0h. The alphanumeric character set is implemented in a 5 x 7 format with one empty column to the right and one empty row above and one below and stored in a 6 x 9 format.

"1" -The space between lines is defined by the value in the vertical space register.

Table 13. Format Character Register Colour Setting.

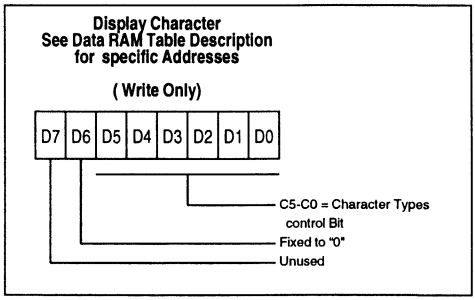
R	G	B	Colour
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

Table 14. Format Character Register Size Setting

GS2	GS1	S	Vertical Height	Horizontal length
0	0	0	18h	6 TDOT
0	0	1	36h	12 TDOT
0	1	0	18h	6 TDOT
0	1	1	54h	18 TDOT
1	0	0	36h	12 TDOT
1	0	1	54h	18 TDOT
1	1	0	36h	12 TDOT
1	1	1	72h	24 TDOT

TDOT= $2/f_{OSC}$

Figure 60. Display Character Register Explanation



D7. This bit is not used.

D6. This bit is fixed to "0".

C5-C0. *Character type.* The 6 character type bits define one of the 64 available character types. These character types are shown on the following pages.

Character Types

The character set is user defined as ROM mask option.

Register and RAM Addressing

The OSD contains seven registers and 80 RAM locations. The seven registers are the Vertical Start Address register, Horizontal Start Address register, Vertical Space register, Horizontal Space register, Background Control register, Global Enable register and Character Bank Select register. The Global Enable register can be written at any time by the ST631xx Core. The other six registers and the RAM can only be read or written to if the global enable is zero.

The six registers and the RAM are located on two pages of the paged memory of the ST631xx MCUs; the Character Bank Select register is located outside the paged memory at address EDh. Each page contains 64 memory locations. This paged memory is at memory locations 00h to 3Fh in the ST631xx memory map. A page of memory is enabled by setting the desired page bit, located in the Data Ram Bank Register, to a one. The page register is location E8h. A one in bit five selects page 5, located on the OSD and a one in bit 6 selects page 6 on the OSD. Table 15 shows the addresses of the OSD registers and RAM.

ON-SCREEN DISPLAY (Continued)

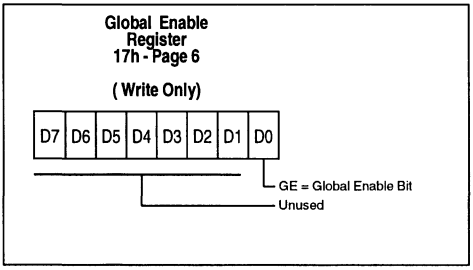
Table 15. OSD Control Registers and Data RAM Addressing

Page	Address	Register or RAM
5	00h - 3Fh	RAM Locations 00h - 3Fh
6	00h - 0Fh	RAM Locations 00h - 0Fh
6	10h	Vertical Start Register
6	11h	Horizontal Start Register
6	12h	Vertical Space Register
6	13h	Horizontal Space Register
6	14h	Background Control Register
6	17h	Global Enable Register
No Page	EDh	Character Bank Select Register

OSD Global Enable Register

This register contains the global enable bit (GE). It is the only register that can be written at any time regardless of the state of the GE bit. It is a write only register.

Figure 61. Global Enable Bit



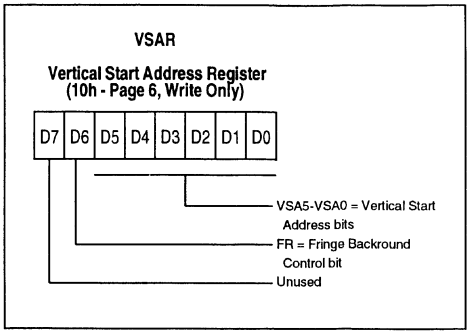
D7-D1. These bits are not used

GE. *Global Enable.* This bit allows the entire display to be turned off.

"0" - The entire display is disabled. The RAM and other registers of the OSD can be accessed by the Core.

"1" - Display of words is controlled by the word enable bits (WE) located in the format or space character. The other registers and RAM cannot be accessed by the Core.

Figure 62. Vertical Start Address Register



D7. This bit is not used

FR. *Fringe Background.* This bit changes the background from a box background to a fringe background. The background is enabled by word as defined by either BG0 or BG1.

"0" - The background is defined to be a box which is 7 x 9 dots.

"1" - The background is defined to be a fringe.

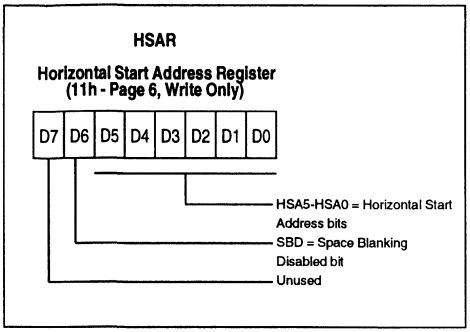
VSA5-VSA0. *Vertical Start Address.* These bits determine the start position of the first line in the vertical direction. The 6 bits can specify 63 display start positions of interval 4h. The first start position will be the fourth line of the display. The vertical start address is defined VSA0 by the following formula.

Vertical Start Address = $4h(2^5(VSA5) + 2^4(VSA4) + 2^3(VSA3) + 2^2(VSA2) + 2^1(VSA1) + 2^0(VSA0))$

The case of all Vertical Start Address bits being zero is illegal.

ON-SCREEN DISPLAY (Continued)

Figure 63. Horizontal Start Address Register



D7. This bit is not used.

SBD. *Space Blanking Disable.* This bit controls whether or not the background is displayed when outputting spaces. If two background colours are used on adjacent words, then the background should not be displayed on spaces in order to make a nice break between colours. If an even background around an area of text is desired, as in a menu, then the background should be displayed when outputting spaces.

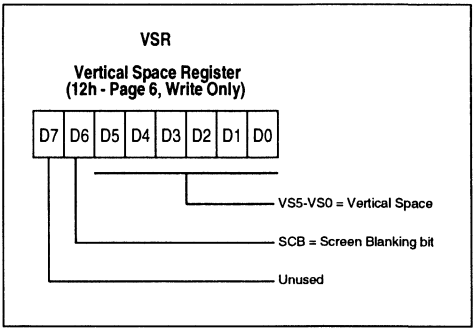
"0" -The background during spaces is controlled by the background enable bits (BG0 and BG1) located in the Background Control register.

"1" -The background is not displayed when outputting spaces.

HSA5, HSA0 - *Horizontal Start Address bits.* These bits determine the start position of the first character in the horizontal direction. The 6 bits can specify 64 display start positions of interval $2/f_{osc}$ or 400ns. The first start position will be at 4.0µs because of the time needed to access RAM and ROM before the first character can be displayed. The horizontal start address is defined by the following formula.

Horizontal Start Address = $2/f_{osc}(10.0 + 2^5(HSA5) + 2^4(HSA4) + 2^3(HSA3) + 2^2(HSA2) + 2^1(HSA1) + 2^0(HSA0))$

Figure 64. Vertical Space Register



D7. This bit is not used

SCB. *Screen Blanking.* This bit allows the entire screen to be blanked.

"0" -The blanking output signal (VBLK) is active only when displaying characters.

"1" -The blanking output signal (VBLK) is always active. Characters in the display RAM are still displayed.

When this bit is set to one, the screen is blanked also without setting the Global Enable bit to one (OSD disabled).

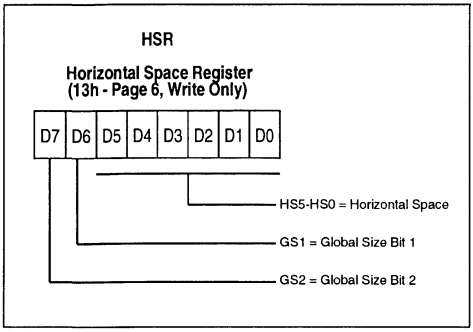
VS5 , VS0. *Vertical Space.* These bits determine the spacing between lines if the Vertical Space Enable bit (VSE) in the format character is one. If VSE is zero there will be no spaces between lines. The Vertical Space bits can specify one of 63 spacing values from 4h to 252h. The space between lines is defined by the following formula.

Space between lines = $4h(2^5(VS5) + 2^4(VS4) + 2^3(VS3) + 2^2(VS2) + 2^1(VS1) + 2^0(VS0))$

The case of all Vertical Start Address bits being zero is illegal.

ON-SCREEN DISPLAY (Continued)

Figure 65. Horizontal Space Register



GS2,GS1. *Global Size.* These bits along with the size bit (S) located in the Character format word specify the character size for each line as defined in Table 16.

Table 16. Horizontal Space Register Size Setting.

GS2	GS1	S	Vertical Height	Horizontal Length
0	0	0	18h	6 TDOT
0	0	1	36h	12 TDOT
0	1	0	18h	6 TDOT
0	1	1	54h	18 TDOT
1	0	0	36h	12 TDOT
1	0	1	54h	18 TDOT
1	1	0	36h	12 TDOT
1	1	1	72h	24 TDOT

Note: TDOT= 2/fOSC

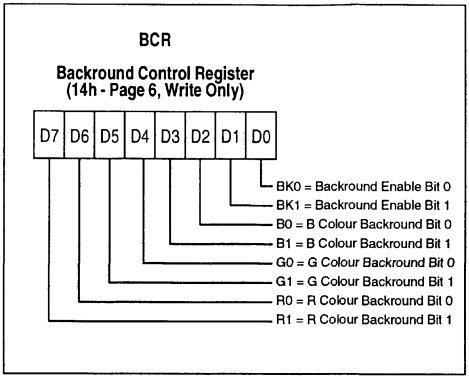
HS5, HS0 . *Horizontal Space* . These bits determine the spacing between words if the Horizontal Space Enable bit (HSE) located in the space character is a one. The space between words is then equal to the width of the space character plus the number of tdots specified by the Horizontal Space bits. The 6 bits can specify one of 64 spacing values ranging from $2/f_{osc}$ to $128/f_{osc}$. The formula is shown below for the smallest size character(18h). If larger size characters are being displayed the spacing between words will increase proportionately. Multiply the value below by 2, 3 or 4 for character sizes of 36h, 54h and 72h respectively.

Space between words (not including the space character)$=2/f_{osc}(1+2^5(HS5)+2^4(HS4)+2^3(HS3)+2^2(HS2)+2^1(HS1)+2^0(HS0))$

Background Control Register

This register sets up two possible backgrounds. The background select bit (BGS) in the format or space character will determine which background is selected for the current word.

Figure 66. Background Control Register



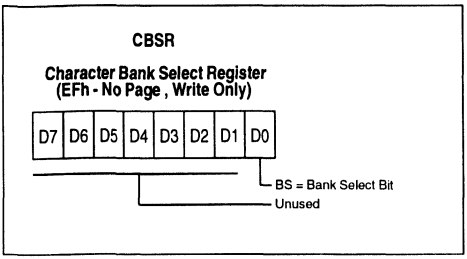
R1,R0,G1,G0,B1,B0. Background Colour. These bits define the colour of the specified background, either background 1 or background 0 as defined in Table 17.

Table 17. Background Register Colour Setting.

RX	GX	BX	Colour
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

ON-SCREEN DISPLAY (Continued)

Figure 67. Character Bank Select Register



BK1,BK0. *Background Enable.*These bits determine if the specified background is enabled or not.

"0" -The following word does not have a background.

"1" -There is a background around the following word.

D7-D1. These bits are not used

BS. *Bank Select.* This bit select the character bank to be used. The lower bank is selected with 0. The value can be modified only when the OSD is OFF (GE=0). No reset value.

OSD Data RAM

The contents of the data RAM can be accessed by the ST631xx MCUs only when the global enable bit (GE) in the Global Enable register is a zero.

The first character in every line is the format character. This character is not displayed. It defines the size of the characters in the line and contains the vertical space enable bit. This character also defines the colour, background and display enable for the first word in the line. Subsequent characters are either spaces or one of the 64 available character types.

The space character defines the colour, background, display enable and horizontal space enable for the following word. Since there are 5 display lines of 15 characters each, the display RAM must contain 5 lines x (15 characters + 1 format character) or 80 locations. The RAM size is 80 locations x 7 bits. The data RAM map is shown inTable 12.

ON-SCREEN DISPLAY (Continued)

Table 12. OSD RAM Map

Column				0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A0				0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
A1				0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
A2				0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
A3				0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Page	A5	A4	LINE																
5	0	0	1	FT	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch
5	0	1	2	FT	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch
5	1	0	3	FT	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch
5	1	1	4	FT	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch
6	0	0	5	FT	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch	Ch
				AVAILABLE SCREEN SPACE															

Notes: FT. The format character required for each line. Characters in columns 1 thru 15 are displayed.
Ch. (Byte) Character (Index into OSD character generator) or space character

Emulator Remarks

There are a few differences between emulator and silicon. For noise reasons, the OSD oscillator pins are not available: the internal oscillator cannot be disabled and replaced by an external coil. In the emulator, the Character Bank Select register can be written also with Global Enable bit set, while this is not allowed in the device.

Application Notes

1 - The OSD character generator is composed of a dual port video ram and some circuitry. It needs two input signals VSYNC and HSYNC to syncronize its dedicated oscillator to the TV picture. It generates 4 output signals, that can be used from the TV set to generate the characters on the screen. For instance, they can be used to feed the SCART plug, providing an adequate buffer to drive the low impedance (75 Ω) of the SCART inputs.

2 - The Core sees the OSD as a number of RAM locations (80) plus a certain number of control registers (6). These 86 locations are mapped in two pages of the dynamic data ram address range (0h..3Fh).
In page 5 (load 20h in the register 0E8h), there are 64 bytes of RAM, the ones of the first 4 rows (16 bytes each row, 15 characters per row maximum, plus an hidden leading format character). In page 6 (load 40h in register 0E8h), the 16 bytes of the fifth row (0..0Fh), and the 6 control registers (10h..14h,17h).

3 - The video RAM is a dual port ram. That means that it can be addressed either from the Core or from the OSD circuitry itself. To reduce the complexity of the circuitry, and thus its cost, some restrictions have been introduced in the use of the OSD.

a. The Core can Only write to any of the 86 locations (either video RAM or control registers).

b. The Core can Only write to any of the leading 85 locations when the OSD oscillator is OFF. Only the last location (control register 17h in page 6) can be addressed at any time. This is the Global Enable Register, which contains only the GE bit. If it is set, the OSD is on, if it is reset the OSD is off.

4 - The timing of the on/off switching of the OSD oscillator is the following:

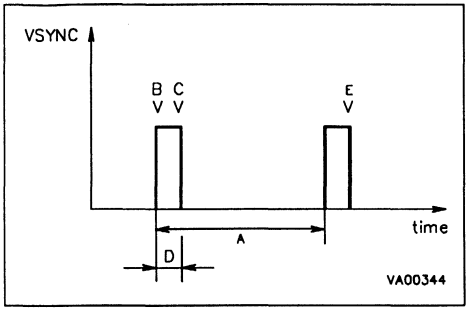
a. GE bit is set. The OSD oscillator will start on the next VSYNC signal.

b. GE bit is reset. The OSD oscillator will be immediately switched off.

ON-SCREEN DISPLAY (Continued)

To avoid a bad visual impression, it is important that the GE bit is set before the end of the flyback time when changing characters. This can be done inside the VSYNC interrupt routine. The following diagram can explain better:

Figure 68. OSD Oscillator ON/OFF Timing



Notes: A - Picture time: 20 mS in PAL/SECAM.
B - VSYNC interrupt, if enabled.
C - Starting of OSD oscillator, if GE = 1.
D - Flyback time.

When modifying the picture display (i.e.: a bar graph for an analog control), it is important that the switching on of the GE bit is done before the the end of the flyback time (D in Figure 68). If the GE bit is set after the end of the flyback time then the OSD will not start until the begining of the next frame. This results in one frame being lost and will result in a Flicker on the screen. One method to be sure to avoid the flicker is to wait for the VSYNC interrupt at the start of the flyback; once the VSYNC interrupt is detected, then the GE bit can be set to zero, the characters changed, and the GE set to one. All this should occur before the end of the flyback time in order not to lose a frame. The correct edge of the interrupt must be chosen.

The VSYNC pin may alternatively be sampled by software in order to know the status; this can be done by reading bit 4 of register E4h; this bit is inverted with respect to the VSYNC pin.

6 - An OSD end of line Bar is present in the ST63P1xx piggyback and ST631xx ROM, EPROM and OTP devices when using the background mode. If this bar is present with software running in the piggybacks then it is also present on the ROM mask version. If the end of line bar is seen to be eliminated by software in the piggyback, then it is also be eliminated in the ROM mask version.

The bar appears at the end of the line in the background mode when the last character is a space character, the first format character is defined with S=0 (size 0)and the backround is not displayed during the space. The bar is the colour of the background defined by the space character. To eliminate the bar:

a. If two backgrounds are used then the bar should be moved off the screen by using large word spaces instead of character spaces. If there are not enough spaces before the end of the line, then the location of the valid characters should be moved so they appear at the end of the line (and hence no bar); positioning can be compensated using the horizontal start register.

b. If only one background is used, then the other background should be transparent in order to eliminate the bar.

7 - The OSD oscillator external network should consist of a capacitor on each of the OSD oscillator pins to ground together with an inductance between pins. The user should select the two capacitors to be the same value (15pF to 25pF each is recommended). The inductance is chosen to give the desired OSD oscillator frequency for the application (typically 56µH).

SOFTWARE DESCRIPTION

The ST631xx software has been designed to fully use the hardware in the most efficient way possible while keeping byte usage to a minimum; in short to provide byte efficient programming capability. The ST631xx Core has the ability to set or clear any register or RAM location bit of the Data space with a single instruction. Furthermore, the program may branch to a selected address depending on the status of any bit of the Data space. The carry bit is stored with the value of the bit when the SET or RES instruction is processed.

Addressing Modes

The ST631xx Core has nine addressing modes which are described in the following paragraphs. The ST631xx Core uses three different address spaces : Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X,Y,V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In the immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In the direct addressing mode, the address of the byte that is processed by the instruction is stored in the location that follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The Core can address the four RAM registers X,Y,V,W (locations 80h, 81h, 82h, 83h) in the short-direct addressing mode. In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of the direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In the extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four less significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) that use the extended addressing mode are able to branch to any address of the 4K bytes Program space.

An extended addressing mode instruction is two-byte long.

Program Counter Relative. The relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations around the address of the relative instruction. If the condition is not true, the instruction that follows the relative instruction is executed. The relative addressing mode instruction is one-byte long. The opcode is obtained in adding the three most significant bits that characterize the kind of the test, one bit that determines whether the branch is a forward (when it is 0) or backward (when it is 1) branch and the four less significant bits that give the span of the branch (0h to Fh) that must be added or subtracted to the address of the relative instruction to obtain the address of the branch.

Bit Direct. In the bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit test and branch instruction is three-byte long. The bit identification and the tested condition are included in the opcode byte. The address of the byte to be tested follows immediately the opcode in the Program space. The third byte is the jump displacement, which is in the range of -126 to +129. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In the indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed by the content of one of the indirect registers, X or Y (80h,81h). The indirect register is selected by the bit 4 of the opcode. A register indirect instruction is one byte long.

Inherent. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

SOFTWARE DESCRIPTION (Continued)

Instruction Set

The ST631xx Core has a set of 40 basic instructions. When these instructions are combined with nine addressing modes, 244 usable opcodes can be obtained. They can be divided into six different types:load/store, arithmetic/logic, conditional branch, control instructions, jump/call, bit manipulation. The following paragraphs describe the different types.

All the instructions within a given type are presented in individual tables.

Load & Store. These instructions use one,two or three bytes in relation with the addressing mode. One operand is the Accumulator for LOAD and the other operand is obtained from data memory using one of the addressing modes.

For Load Immediate one operand can be any of the 256 data space bytes while the other is always immediate data. See Table 13.

Table 13. Load & Store Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
LD A, X	Short Direct	1	4	Δ	*
LD A, Y	Short Direct	1	4	Δ	*
LD A, V	Short Direct	1	4	Δ	*
LD A, W	Short Direct	1	4	Δ	*
LD X, A	Short Direct	1	4	Δ	*
LD Y, A	Short Direct	1	4	Δ	*
LD V, A	Short Direct	1	4	Δ	*
LD W, A	Short Direct	1	4	Δ	*
LD A, rr	Direct	2	4	Δ	*
LD rr, A	Direct	2	4	Δ	*
LD A, (X)	Indirect	1	4	Δ	*
LD A, (Y)	Indirect	1	4	Δ	*
LD (X), A	Indirect	1	4	Δ	*
LD (Y), A	Indirect	1	4	Δ	*
LDI A, #N	Immediate	2	4	Δ	*
LDI rr, #N	Immediate	3	4	*	*

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
#. Immediate data (stored in ROM memory)
rr. Data space register
Δ. Affected
*. Not Affected

SOFTWARE DESCRIPTION (Continued)

Arithmetic and Logic. These instructions are used to perform the arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while the other can be either a data space memory content or an immediate value in relation with the addressing mode. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator. See Table 14.

Table 14. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	*
AND A, (Y)	Indirect	1	4	Δ	*
AND A, rr	Direct	2	4	Δ	*
ANDI A, #N	Immediate	2	4	Δ	*
CLR A	Short Direct	2	4	Δ	Δ
CLR rr	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y. Indirect Register Pointers, V & W Short Direct Registers
#. Immediate data (stored in ROM memory)
rr. Data space register

Δ. Affected
*. Not Affected

SOFTWARE DESCRIPTION (Continued)

Conditional Branch. The branch instructions achieve a branch in the program when the selected condition is met. See Table 15.

Bit Manipulation Instructions. These instructions can handle any bit in data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations. See Table 16.

Control Instructions. The control instructions control the MCU operations during program execution. See Table 17.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutines call inside the whole program space. Refer to Table 18.

Table 15. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b. 3-bit address
e. 5 bit signed displacement in the range -15 to +16
ee. 8 bit signed displacement in the range -126 to +129
rr. Data space register
Δ. Affected
*. Not Affected

Table 16. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b. 3-bit address;
rr. Data space register;
*. Not Affected

Table 17. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated and a WAIT is automatically executed instead of a STOP if the hardware activated watchdog function is selected.

Δ. Affected
*. Not Affected

Table 18. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc. 12-bit address;
*. Not Affected

SOFTWARE DESCRIPTION (Continued)

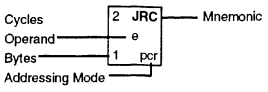
Opcode Map Summary. The following table contains an opcode map for the instructions used on the MCU.

Low / Hi	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Low / Hi
0 0000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 pcr	4 LD a,(y) 1 ind	0 0000
1 0001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC x 1 sd	2 JRC e 1 prc	4 LDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 JRZ e 1 pcr	4 DEC x 1 sd	2 JRC e 1 pcr	4 LD a,rr 2 dir	1 0001
2 0010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b4,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 CP a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b4,rr 2 b.d	2 JRZ e 1 pcr	4 COM a 1 inh	2 JRC e 1 pcr	4 CP a,(y) 1 ind	2 0010
3 0011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b4,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,x 1 sd	2 JRC e 1 prc	4 CPI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b4,rr 2 b.d.	2 JRZ e 1 pcr	4 LD x,a 1 sd	2 JRC e 1 pcr	4 CP a,rr 2 dir	3 0011
4 0100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b2,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 ADD a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b2,rr 2 b.d	2 JRZ e 1 pcr	2 RETI 1 inh	2 JRC e 1 pcr	4 ADD a,(y) 1 ind	4 0100
5 0101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b2,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC y 1 sd	2 JRC e 1 prc	4 ADDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b2,rr 2 b.d	2 JRZ e 1 pcr	4 DEC y 1 sd	2 JRC e 1 pcr	4 ADD a,rr 2 dir	5 0101
6 0110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b6,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 INC (x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b6,rr 2 b.d	2 JRZ e 1 pcr	2 STOP 1 inh	2 JRC e 1 pcr	4 INC (y) 1 ind	6 0110
7 0111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b6,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,y 1 sd	2 JRC e 1 prc	#	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b6,rr 2 b.d	2 JRZ e 1 pcr	4 LD y,a 1 sd	2 JRC e 1 pcr	4 INC rr 2 dir	7 0111
8 1000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b1,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (x),a 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b1,rr 2 b.d	2 JRZ e 1 pcr	#	2 JRC e 1 pcr	4 LD (y),a 1 ind	8 1000
9 1001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b1,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC v 1 sd	2 JRC e 1 prc	#	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b1,rr 2 b.d	2 JRZ e 1 pcr	4 DEC v 1 sd	2 JRC e 1 pcr	4 LD rr,a 2 dir	9 1001
A 1010	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b5,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 AND a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b5,rr 2 b.d	2 JRZ e 1 pcr	4 RLC a 1 inh	2 JRC e 1 pcr	4 AND a,(y) 1 ind	A 1010
B 1011	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b5,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,v 1 sd	2 JRC e 1 prc	4 ANDI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b5,rr 2 b.d	2 JRZ e 1 pcr	4 LD v,a 1 sd	2 JRC e 1 pcr	4 AND a,rr 2 dir	B 1011
C 1100	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b3,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 SUB a,(x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b3,rr 2 b.d	2 JRZ e 1 pcr	2 RET 1 inh	2 JRC e 1 pcr	4 SUB a,(y) 1 ind	C 1100
D 1101	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b3,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC w 1 sd	2 JRC e 1 prc	4 SUBI a,nn 2 imm	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b3,rr 2 b.d	2 JRZ e 1 pcr	4 DEC w 1 sd	2 JRC e 1 pcr	4 SUB a,rr 2 dir	D 1101
E 1110	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b7,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 DEC (x) 1 ind	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b7,rr 2 b.d	2 JRZ e 1 pcr	2 WAIT 1 inh	2 JRC e 1 pcr	4 DEC (y) 1 ind	E 1110
F 1111	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b7,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,w 1 sd	2 JRC e 1 prc	#	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b7,rr 2 b.d	2 JRZ e 1 pcr	4 LD w,a 1 sd	2 JRC e 1 pcr	4 DEC rr 2 dir	F 1111

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:
# Indicates Illegal Instructions
e 5 Bit Displacement
b 3 Bit Address
rr1byte dataspace address
nn 1 byte immediate data
abc 12 bit address
ee 8 bit Displacement



ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from :

$$Tj = T_A + PD \times RthJA$$

Where : T_A = Ambient Temperature.

RthJA = Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint = $I_{DD} \times V_{DD}$ (chip internal power).

Pport = Port power dissipation (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage (AFC IN)	V_{SS} - 0.3 to +13	V
V_I	Input Voltage (Other Inputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
V_O	Output Voltage (PA4-PA7, PC4-PC7, DA0-DA5)	V_{SS} - 0.3 to +13	V
V_O	Output Voltage (Other Outputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
I_O	Current Drain per Pin Excluding V_{DD}, V_{SS}, PA6, PA7	± 10	mA
I_O	Current Drain per Pin (PA6, PA7)	± 50	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	150	mA
Tj	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Note : Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device . This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

THERMAL CHARACTERISTIC

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
RthJA	Thermal Resistance	PDIP40 PDIP28			38 55	°C/W

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature		0		70	°C
V_{DD}	Operating Supply Voltage		4.5	5.0	6.0	V
f_{OSC}	Oscillator Frequency RUN & WAIT Modes			8	8.1	MHz
f_{OSDOSC}	On-screen Display Oscillator Frequency				8.0	MHz

EEPROM INFORMATION

The ST631xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = 0 to +70°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	All I/O Pins, KBY0-2			0.3xV_{DD}	V
V_{IH}	Input High Level Voltage	All I/O Pins, KBY0-2	0.75xV_{DD}			V
V_{HYS}	Hysteresis Voltage(1)	All I/O Pins, KBY0-2 V_{DD} = 5V		1.0		V
V_{OL}	Low Level Output Voltage	Port B/C, DA0-3, BSW0-3, OUT1, VS, OSD Outputs, V_{DD} = 4.5V I_{OL} = 1.6mA I_{OL} = 5.0mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	Port A V_{DD} = 4.5V I_{OL}= 3.2mA I_{OL}= 30mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	OSDOSCout, OSCout V_{DD} = 4.5V I_{OL}= 0.1mA			 0.4	 V
V_{OH}	High Level Output Voltage	Port B/C(2), VS V_{DD} = 4.5V I_{OH} = – 1.6mA	 4.1			 V
V_{OH}	High Level Output Voltage	OSDOSCout, OSCout, V_{DD} = 4.5V I_{OL}= – 0.1mA	 4.1			 V
I_{PU}	Input Pull Up Current Input Mode with Pull-up	Port B/C, KBY0-2 V_{IN}= V_{SS}(2)	– 100	– 50	– 25	mA
I_{IL} I_{IH}	Input Leakage Current	OSCin V_{IN}= V_{SS} V_{IN}= V_{DD}	– 10 0.1	– 1 1	– 0.1 10	µA
I_{IL} I_{IH}	Input Leakage Current	All I/O Input Mode no Pull-up OSDOSCin V_{IN}= V_{DD} or V_{SS}	– 10		10	µA
I_{IL} I_{IH}	Input Leakage Current	Reset Pin with Pull-up V_{IN}= V_{SS}	– 50	– 30	– 10	µA
I_{IL} I_{IH}	Input Leakage Current	AFC Pin V_{IH}= V_{DD} V_{IL}= V_{SS} V_{IH}= 12.0V	 – 1		 1 40	 µA
I_{OH}	Output Leakage Current	Port A, DA0-3, BSW0-3 OUT1, OSDout V_{OH} = V_{DD}			10	µA
I_{OH}	Output Leakage Current High Voltage	Port A, DA0-3, BSW0-3 OUT1 V_{OH} = 12V			40	µA

DC ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
I_{DD}	Supply Current RUN Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6.0V		6	16	mA
I_{DD}	Supply Current WAIT Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6V		3	10	mA
V_{ON}	Reset Trigger Level ON	RESET Pin			$0.3xV_{DD}$	V
V_{OFF}	Reset Trigger Level OFF	RESET Pin	$0.8xV_{DD}$			V
V_{TA}	Input Level Absolute Tolerance	A/D AFC Pin V_{DD} = 5V			±200	mV
V_{TR}	Input Level Relatice Tolerance	A/D AFC Pin Relative to other levels V_{DD} = 5V			±100	mV

Notes:
1. Not 100% Tested
2. Input pull-up option only

AC ELECTRICAL CHARACTERISTICS

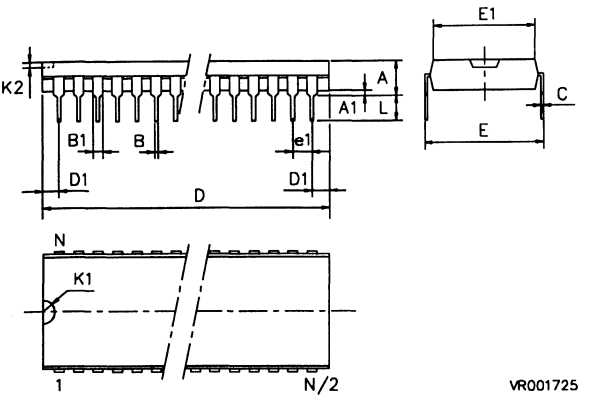
(T_A = 0 to +70°C, f_{OSC}=8MHz, V_{DD}=4.5 to 6.0V unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{WRES}	Minimum Pulse Width	RESET Pin	125			ns
tO_{HL}	High to Low Transition Time	PA6, PA7 V_{DD} = 5V, CL = 1000pF (2)		100		ns
tO_{HL}	High to Low Transition Time	DA0-DA5, PB0-PB6, OSD Outputs, PC0-PC7, V_{DD} = 5V, CL = 100pF		20		ns
tO_{LH}	Low to High Transition Time	PB0-PB6, PA0-PA3, OSD Outputs, PC0-PC3 V_{DD} = 5V, CL = 100pF		20		ns
tO_H	Data HOLD Time SPI after clock goes low I^2CBUS/S-BUS Only		175			ns
f_{DA}	D/A Converter Repetition Frequency(1)			31.25		kHz
f_{SIO}	SIC Baud Rate(1)			62.50		kHz
t_{WEE}	EEPROM Write Time	T_A = 25°C One Byte		5	10	ms
Endurance	EEPROM WRITE/ERASE Cycles	Q_A L_{OT} Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention (4)	T_A = 25°C	10			years
C_{IN}	Input Capacitance (3)	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance (3)	All outputs Pins			10	pF
COSCin, COSCout	Oscillator Pins Internal Capacitance(3)			5		pF
COSDin, COSDout	OSD Oscillator External Capacitance	Recommended	15		25	pF

Notes:
1. A clock other than 8 MHz will affect the frequency response of those peripherals (D/A, 62.5kHz and SPI) whose clock is derived from the system clock.
2. The rise and fall times of PORT A have been reduced in order to avoid current spikes while maintaining a high drive capability
3. Not 100% Tested
4. Based on extrapolated data

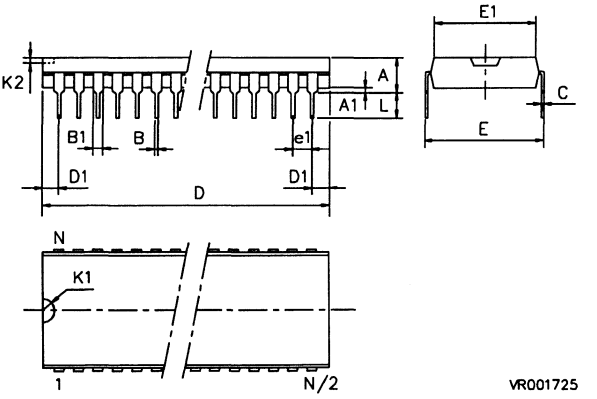
PACKAGE MECHANICAL DATA

Figure 69. ST631xx 40 Pin Plastic Dual-In-line Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.2		4.8	0.086		0.189
A1	0.51		1.77	0.010		0.069
B	0.38		0.58	0.015		0.023
B1	0.97		1.52	0.055		0.065
C	0.20		0.30	0.008		0.009
D	50.30		52.22	1.980		20.560
D1	–	–	–	–	–	–
E		15.2			0.600	
E1		12.9			0.508	
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	3.18		4.44	1.25		0.174
e1		2.54			0.10	
Number of Pins						
N	40					

Figure 70. ST631xx 28-Pin Dual-In-line Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.2		4.8	0.086		0.189
A1	0.51		1.77	0.010		0.069
B	0.38		0.58	0.015		0.023
B1	0.97		1.52	0.055		0.065
C	0.20		0.30	0.008		0.009
D	35.06		36.22	1.400		1.425
D1	–	–	–	–	–	–
E		15.2			0.600	
E1		12.9			0.508	
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	3.18		4.44	1.25		0.174
e1		2.54			0.10	
Number of Pins						
N	28					

ORDERING INFORMATION

The following chapter deals with the procedure for transfer the Program/Data ROM codes to SGS-THOMSON.

Communication of the ROM Codes. To communicate the contents of Program /Data ROM memories to SGS-THOMSON, the customer has to send a 5" Diskette with:

- one file in INTEL INTELLEC 8/MDS FORMAT for the PROGRAM Memory
- one file in INTEL INTELLEC 8/MDS FORMAT for the ODD and EVEN ODD OSD Characters
- one file in INTEL INTELLEC 8/MDS FORMAT for the EEPROM initial content (this file is optional)
- a filled Option List form as described in the OPTION LIST paragraph.

The program ROM should respect the ROM Memory Map as in Table 19.

The ROM code must be generated with ST6 assembler. Before programming the EPROM, the buffer of the EPROM programmer must be filled with FFh.

Table 19. ROM Memory Map

ROM Page	Device Address	EPROM Address (1)	Description
Page 0	0000h-007Fh 0080h-07FFh	0000h-007Fh 0080h-07FFh	Reserved User ROM
Page 1 "STATIC"	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
Page 2	0000h-000Fh 0010h-07FFh	1000h-100Fh 1010h-17FFh	Reserved User ROM
Page 3	0000h-000Fh 0010h-07FFh	1800h-180Fh 1810h-1FFFh	Reserved user ROM

Note 1. EPROM addresses are related to the use of ST63E1xx EPROM emulation devices.

ORDERING INFORMATION (Continued)

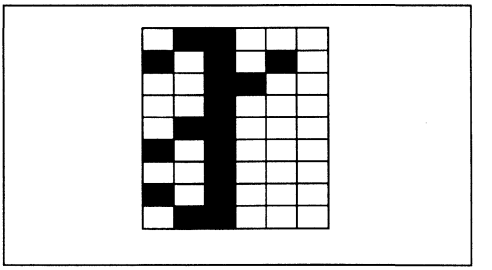
Customer EEPROM Initial Contents: Format

a. The content should be written into an INTEL INTELLEC format file.

b. Undefined or don't care bytes should have the content FFh.

OSD Test Character. IN ORDER TO ALLOW THE TESTING OF THE ON-CHIP OSD MACROCELL THE FOLLOWING CHARACTER MUST BE PROVIDED AT THE FIXED 3Fh (63) POSITION OF THE SECOND OSD BANK.

Figure 71. OSD Test Character



Listing Generation & Verification. When SGS-THOMSON receives the files, a computer listing is generated from them. This listing refers extractly to the mask that will be used to produce the microcontroller. Then the listing is returned to the customer that must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed list constitutes a part of the contractual agreement for the creation of the customer mask. SGS-THOMSON sales organization will provide detailed information on contractual points.

ORDERING INFORMATION TABLE

Sales Type	ROM/EEPROM	Temperature Range	Package
ST63140B1/XX	8K (EPROM) / 128 Bytes	0 to + 70 ° C	PDIP28
ST63142B1/XX		0 to + 70 ° C	PDIP28
ST63126B1/XX		0 to + 70 ° C	PDIP40
ST63156B1/XX		0 to + 70 ° C	PDIP40

Note. /XX Is the ROM Code idebtifier that is allocated by SGS-THOMSON after receipt of all required options and the related ROM file

ST631xx MICROCONTROLLER OPTION LIST

Customer: . . . . . .
Address: . . . . . .
Contact: . . . . . .
Phone No: . . . . . .
Reference: . . . . . .

Device [ ] ST63140 [ ] ST63142 [ ] ST63126 [ ] ST63156
Temperature Range 0 to 70°C

For marking one line with 12 characters maximum is possible
Special Marking [ ] No
[ ] Yes Line1 " _ _ _ _ _ _ _ _ _ _ _ _ "

Letters, digits, ' . ', ' - ', ' / ' and spaces only
the default marking is equivalent to the sales type only (part number).

OSD POLARITY OPTIONS (Put a cross on selected item) :

	POSITIVE	NEGATIVE
VSYNC,HSYNC	[ ]	[ ]
R,G,B	[ ]	[ ]
BLANK	[ ]	[ ]

CHECK LIST:

	YES	NO
ROM CODE	[ ]	[ ]
OSD Code: ODD & EVEN	[ ]	[ ]
EEPROM Code (if Desired)	[ ]	[ ]

Signature .....

Date .....

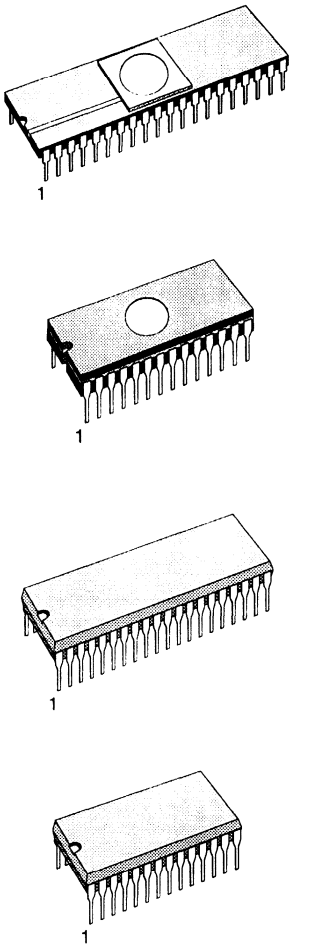
SGS-THOMSON MICROELECTRONICS

ST63E140/T140, E142/T142
ST63E126/T126, E156/T156

8-BIT EPROM HCMOS MCUs FOR TV FREQUENCY & VOLTAGE SYNTHESIS WITH OSD

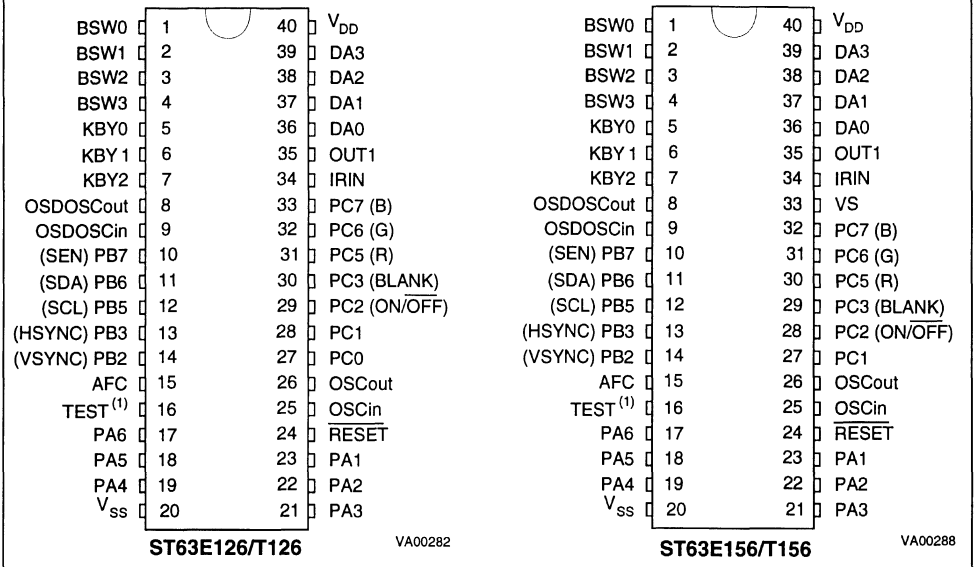
PRELIMINARY DATA

- 4.5 to 6V operating Range
- 8MHz Maximum Clock Frequency
- User Program EPROM: 7948 bytes
- Reserved Test EPROM: 244 bytes
- Data EPROM: user selectable size
- Data RAM: 256 bytes
- Data EEPROM: 128 bytes
- 40-Pin Dual in Line Package for the ST63x126, x156
- 28-Pin Dual in Line Package for the ST63x140, x142
- Up to 18 software programmable general purpose Inputs/Outputs, including 8 direct LED driving Outputs
- 3 Inputs for keyboard scan (KBY0-2)
- Up to 4 high voltage outputs (BSW0-3)
- Two Timers each including an 8-bit counter with a 7-bit programmable prescaler
- Digital Watchdog Function
- Serial Peripheral Interface (SPI) supporting S-BUS/ I^2C BUS and standard serial protocols
- Up to Four 6-bit PWM D/A Converters
- 62.5kHz Output pin
- 14 bit counter for voltage synthesis tuning (ST63156, ST63140)
- AFC A/D converter with 0.5V resolution
- Four interrupt vectors (IRIN/NMI, Timer 1 & 2, VSYNC.)
- On-chip clock oscillator
- 5 Lines by 15 Characters On-Screen Display Generator with 128 Characters (2 banks)
- These EPROM and OTP versions are fully pin to pin compatible with their respective ROM versions
- The development tool of the ST631xx microcontrollers consists of the ST63TVS-EMU emulation and development system to be connected via a standard RS232 serial line to an MS-DOS Personal Computer.
- EPROM programming board ST63E1XX-EPB



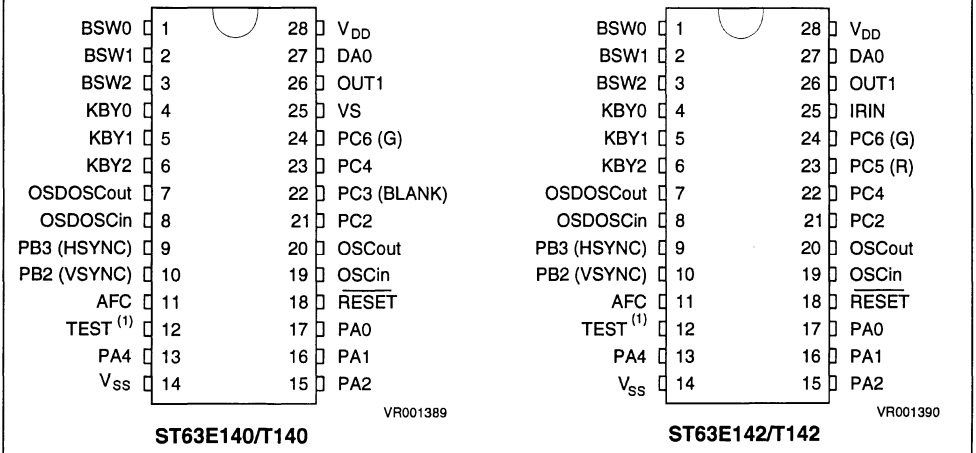
(Ordering Information at the end of the datasheet)

Figure 1. ST63E126/T126, E156/T156 Pin Configuration



Note 1. This pin is also the VPP input for EPROM based devices

Figure 2. ST63E140/T140, E142/T142 Pin Configuration



Note 1. This pin is also the VPP input for EPROM based devices

GENERAL DESCRIPTION

The ST63E140/T140, E142/T142, E126/T126, E156/T156 microcontrollers are members of the 8-bit HCMOS ST631xx family, a series of devices specially oriented to TV applications. Different peripheral configurations are available to give the maximum application and cost flexibility. All ST631xx members are based on a building block approach: a common core is surrounded by a combination of on-chip peripherals (macrocells) available from a standard library. These peripherals are designed with the same Core technology providing full compatibility and short design time. Many of these macrocells are specially dedicated to TV applications.

The macrocells of the ST631xx family are: two Timer peripherals each including an 8-bit counter with a 7-bit software programmable prescaler (Timer), a digital hardware activated watchdog function (DHWD), a 14-bit voltage synthesis tuning peripheral, a Serial Peripheral Interface (SPI), up to four 6-bit PWM D/A converters, an AFC A/D converter with 0.5V resolution, an on-screen display (OSD) with 15 characters per line and 128 characters (in two banks each of 64 characters). In addition the following Memory resources are available: program EPROM (8K), data RAM (256 bytes), EEPROM (128 bytes).

Refer to pin configuration figures and to ST631xx device summary (Table 1) for the definition of ST631xx family members and a summary of differences among the different types.

Figure 3. ST631xx family Block Diagram

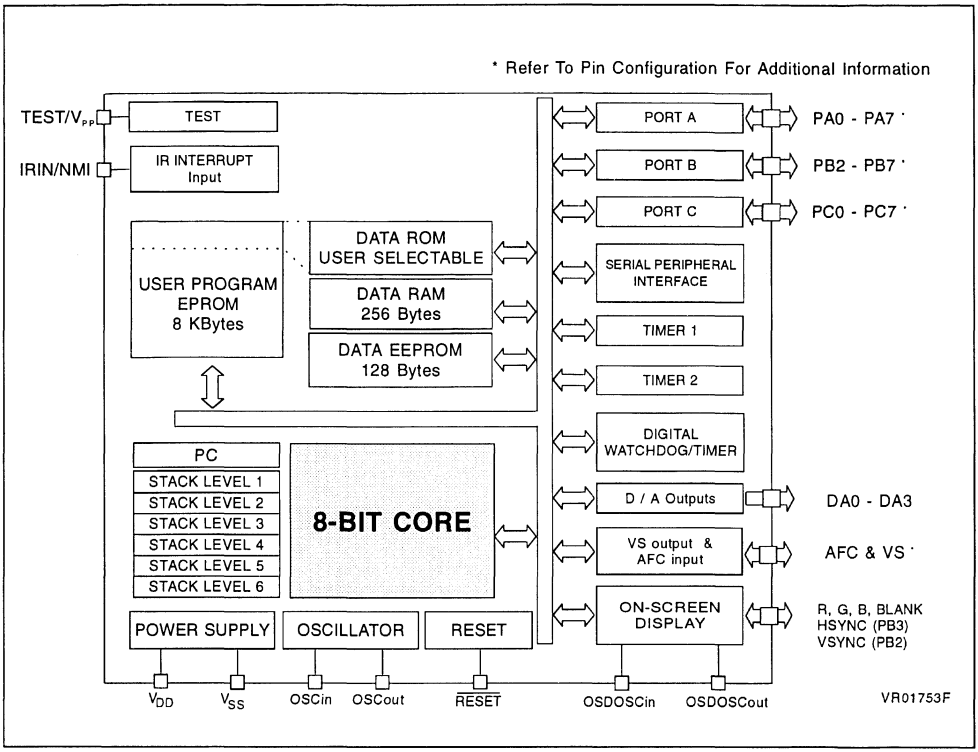


Table 1. Device Summary

DEVICE	EPROM (Bytes)	OTP ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	I/O	KBY I/O	BSW OUT	AFC	VS	D/A	PACK.	TARGET ROM DEVICES
ST63E140	8K		256	128	6	3	3	YES	YES	1	PDIP28	ST63140
ST63T140		8K	256	128	6	3	3	YES	YES	1	PDIP28	ST63140
ST63E142	8K		256	128	6	3	3	YES	NO	1	PDIP28	ST63142
ST631T42		8K	256	128	6	3	3	YES	NO	1	PDIP28	ST63142
ST63E126	8K		256	128	12	3	4	YES	NO	4	PDIP40	ST63126
ST63T126		8K	256	128	12	3	4	YES	NO	4	PDIP40	ST63126
ST63E156	8K		256	128	11	3	4	YES	YES	4	PDIP40	ST63156
ST63T156		8K	256	128	11	3	4	YES	YES	4	PDIP40	ST63156

PIN DESCRIPTION

V_{DD} and V_{SS}. Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is the ground connection.

OSCin, OSCout. These pins are internally connected to the on-chip oscillator circuit. A quartz crystal or a ceramic resonator can be connected between these two pins in order to allow the correct operation of the MCU with various stability/cost trade-offs. The OSCin pin is the input pin, the OSCout pin is the output pin.

$\overline{\text{RESET}}$. The active low $\overline{\text{RESET}}$ pin is used to start the microcontroller to the beginning of its program.

TEST/V_{PP}. The TEST pin must be held at V_{SS} for normal operation. If this pin is connected to a 12.5V level during the reset phase, the EPROM programming mode is entered.

Caution. Exceeding 13V on TEST/V_{PP} pin will permanently damaged the device.

PA0-PA7. These 8 lines are organized as one I/O port (A). Each line may be configured as either an input or as an output under software control of the data direction register. Port A has an open-drain (12V drive) output configuration with direct LED driving capability (30mA, 1V).

PB2-PB3, PB5-PB7. These lines are organized as one I/O port (B). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register. PB2-PB3 have a push-pull configuration in output mode while PB5-PB7 are open-drain (5V drive).

PB2 and PB3 lines are connected to the VSYNC and HSYNC control signals of the OSD cell; to provide the right signals to the OSD these I/O lines should be programmed in input mode and the user can read "on the fly" the state of VSYNC and HSYNC signals. PB2 is also connected with the VSYNC Interrupt. The active polarity of VSYNC Interrupt signal is software controlled. The active polarity of these synchronization input pins to the OSD macrocell can be selected by the user as ROM mask option. If the device is specified to have negative logic inputs, then when these signals are low the OSD oscillator stops. If the device is specified to have positive logic inputs, then when these signals are high the OSD oscillator stops.

PB5, PB6 and PB7 lines, when in output modes, are "ANDed" with the SPI control signals. PB5 is connected with the SPI clock signal (SCL), PB6 with the SPI data signal (SDA) while PB7 is connected with SPI enable signal (SEN).

PC0-PC7. These 8 lines are organized as one I/O port (C). Each line may be configured as either an input with or without internal pull-up resistor or as an output under software control of the data direction register. PC0-PC2, PC4 have a push-pull configuration in output mode while PC3, PC5-PC7 (OSD signals) are open-drain (5V drive). PC3, PC5 , PC6 and PC7 lines when in output mode are "ANDed" with the character and blank signals of the OSD cell. PC3 is connected with the OSD BLANK signal, PC5, PC6 and PC7 with the OSD R, G and B signals. The active polarity of these signals can be selected by the user as ROM mask option. PC2 is also used as TV set ON-OFF switch (5V drive).

DA0-DA3. These pins are the four PWM D/A outputs (with 32kHz repetition) of the 6-bit on-chip D/A converters. The PWM function can be disabled by software and these lines can be used as general purpose open-drain outputs (12V drive).

IRIN. This pin is the external NMI of the MCU.

OUT1. This pin is the 62.5kHz output specially suited to drive multi-standard chroma processors. This function can be disabled by software and the pin can be used as general purpose open-drain output (12V drive).

BSW0-BSW3. These output pins can be used to select up to 4 tuning bands. These lines are configured as open-drain outputs (12V drive).

KBY0-KBY2. These pins are input only and can be used for keyboard scan. They have CMOS threshold levels with Schmitt Trigger and on-chip 100kΩ pull-up resistors.

AFC. This is the input of the on-chip 10 level comparator that can be used to implement the AFC function. This pin is an high impedance input able to withstand signals with a peak amplitude up to 12V.

OSDOSCin, OSDOSCout. These are the On Screen Display oscillator terminals. An oscillation capacitor and coil network have to be connected to provide the right signal to the OSD.

VS. This is the output pin of the on-chip 14-bit voltage synthesis tuning cell (VS). The tuning signal present at this pin gives an approximate resolution of 40kHz per step over the UHF band. This line is a push-pull output with standard drive (ST63140, ST63156 only).

PIN DESCRIPTION (Continued)

Table 2. Pin Summary

Pin Function	Description
DA0 to DA3	Output, Open-Drain, 12V
BSW0 to BSW3	Output, Open-Drain, 12V
IRIN	Input, Resistive Bias, Schmitt Trigger
AFC	Input, High Impedance, 12V
OUT1	Output, Open-Drain, 12V
KBY0 to KBY2	Input, Pull-up, Schmitt Trigger
R,G,B, BLANK	Output, Open-Drain, 5V
HSYNC, VSYNC	Input, Pull-up, Schmitt Trigger
OSDOSCin	Input, High Impedance
OSDOSCout	Output, Push-Pull
TEST/V_{PP}	Input, Pull-Down
OSCin	Input, Resistive Bias, Schmitt Trigger to Reset Logic Only
OSCout	Output, Push-Pull
RESET	Input, Pull-up, Schmitt Trigger Input
VS	Output, Push-Pull
PA0-PA6	I/O, Open-Drain, 12V, No Input Pull-up, Schmitt Trigger, High Drive
PB2-PB3, PB5-PB7	I/O, Push-Pull, 5V, Input Pull-up, Schmitt Trigger
PB5-PB7	I/O, Open-Drain, 5V, Input Pull-up, Schmitt Trigger
PC0-PC2, PC4	I/O, Push-Pull, 5V, Input Pull-up, Schmitt Trigger
PC3, PC5-PC7	I/O, Open-Drain, 5V, Input Pull-up, Schmitt Trigger
V_{DD}, V_{SS}	Power Supply Pins

MEMORY SPACE

Table 3. EPROM Memory Map

EPROM Page	Device Address	EPROM Address	Description
Page 0	0000h-007Fh 0080h-07FFh	0000h-007Fh 0080h-07FFh	Reserved User ROM
Page 1 "STATIC"	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	0800h-0F9Fh 0FA0h-0FEFh 0FF0h-0FF7h 0FF8h-0FFBh 0FFCh-0FFDh 0FFEh-0FFFh	User ROM Reserved Interrupt Vectors Reserved NMI Vector Reset Vector
Page 2	0000h-000Fh 0010h-07FFh	1000h-100Fh 1010h-17FFh	Reserved User ROM
Page 3	0000h-000Fh 0010h-07FFh	1800h-180Fh 1810h-1FFFh	Reserved user ROM

EPROM/OTP DESCRIPTION.

The ST63E1xx represents the generic part number for the EPROM versions of the ST63140, 42, 26, 56 ROM products. They are intended for use during the development of an application, and for pre-production and small volume production.
The ST63T1xx OTP have the same characteristics.
They both include EPROM memory instead of the ROM memory of the ST631xx, and so the program and constants of the program can be easily modified by the user with the ST63E1XX EPROM Programming Board from SGS-THOMSON.

The ROM mask options of the ST631xx for OSD polarities (HSYNC, VSYNC, R, G, B, BLANK) are emulated with an **EPROM OPTION BYTE**. This is programmed by the SGS-THOMSON EPROM programming board and its associated software.

The EPROM Option Byte content will define the OSD options as follows :

7							0
Opt7	Opt6	Opt5	Opt4	Opt3	Opt 2	Opt 1	Opt 0

Opt7-Opt6. Device specific bits (1)

Opt5 : This bit define the BLANK polarity,
if 0 the polarity will be negative
if 1 the polarity will be positive..

Opt 4 : This bit define the RGB polarity,
if 0 the polarity will be negative
if 1 the polarity will be positive..

Opt 3 : This bit define the OSD H/Vsync polarity,
if 0 the polarity will be negative
if 1 the polarity will be positive.

Opt2-Opt0. Device specific bits (1)

Note 1. Device specific bits. These reserved bits must be programmed according to the following table for their relevant device.

Sales Type	Opt7-Opt6	Opt2-Opt0
ST63E140/T140	0 0	1 0 0
ST63E142/T142	0 0	1 0 1
ST63E126/T126	0 0	1 0 1
ST63E156/T156	0 0	1 0 1

From a user point of view (with the following exceptions) the ST63E1xx,T1xx products have exactly the same software and hardware features of the ROM version. An additional mode is used to configure the part for programming of the EPROM, this is set by a +12.5V voltage applied to the TEST/V_{PP} pin. The programming of the ST63E1xx,T1xx is described in the User Manual of the EPROM Programming board.

On the ST63E1xx, all the 7948 bytes of PROGRAM memory are available for the user, as all the EPROM memory can be erased by exposure to UV light. On the ST63T1xx (OTP device) a reserved area for test purposes exists, as for the ST631xx ROM device. In order to avoid any discrepancy between program functionality when using the EPROM, OTP and ROM it is recommended NOT TO USE THESE RESERVED AREAS, even when using the ST63E1xx. The Table 3 is a summary of the EPROM/ROM Map and its reserved area.

THE READER IS ASKED TO REFER TO THE DATASHEET OF THE ST631xx ROM-BASED DEVICE FOR FURTHER DETAILS.

EPROM ERASING

The EPROM of the windowed package of the ST63E1xx may be erased by exposure to Ultra Violet light.

The erasure characteristic of the ST63E1xx EPROM is such that erasure begins when the memory is exposed to light with wave lengths shorter than approximately 4000Å. It should be noted that sunlight and some types of fluorescent lamps have wavelengths in the range 3000-4000Å. It is thus recommended that the window of the ST63E1xx package be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.
The recommended erasure procedure of the ST63E1xx EPROM is exposure to short wave ultraviolet light which has wavelength 2537Å. The integrated dose (i.e. UV intensity x exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm^2 power rating. The ST63E1xx should be placed within 2.5 cm (1 inch) of the lamp tubes during erasure.

ABSOLUTE MAXIMUM RATINGS

This product contains devices to protect the inputs against damage due to high static voltages, however it is advised to take normal precaution to avoid application of any voltage higher than maximum rated voltages.

For proper operation it is recommended that V_I and V_O must be higher than V_{SS} and smaller than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriated logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, Tj, in Celsius can be obtained from :

$T_j = T_A + PD \times RthJA$

Where : T_A = Ambient Temperature.

RthJA = Package thermal resistance (junction-to ambient).

PD = Pint + Pport.

Pint = $I_{DD} \times V_{DD}$ (chip internal power).

Pport = Port power dissipation (determinated by the user).

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	-0.3 to 7.0	V
V_I	Input Voltage (AFC IN)	V_{SS} - 0.3 to +13	V
V_I	Input Voltage (Other Inputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
V_O	Output Voltage (PA4-PA7, PC4-PC7, DA0-DA5)	V_{SS} - 0.3 to +13	V
V_O	Output Voltage (Other Outputs)	V_{SS} - 0.3 to V_{DD} +0.3	V
V_{PP}	EPROM programming Voltage	-0.3 to 13.0	V
I_O	Current Drain per Pin Excluding V_{DD}, V_{SS}, PA6, PA7	± 10	mA
I_O	Current Drain per Pin (PA6, PA7)	± 50	mA
IV_{DD}	Total Current into V_{DD} (source)	50	mA
IV_{SS}	Total Current out of V_{SS} (sink)	150	mA
Tj	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-60 to 150	°C

Note : Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device . This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature		0		70	°C
V_{DD}	Operating Supply Voltage		4.5	5.0	6.0	V
f_{OSC}	Oscillator Frequency RUN & WAIT Modes			8	8.1	MHz
f_{OSDOSC}	On-screen Display Oscillator Frequency				8.0	MHz

EEPROM INFORMATION

The ST631xx EEPROM single poly process has been specially developed to achieve 300.000 Write/Erase cycles and a 10 years data retention.

DC ELECTRICAL CHARACTERISTICS

(T_A = 0 to +70°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IL}	Input Low Level Voltage	All I/O Pins, KBY0-2			0.3xV_{DD}	V
V_{IH}	Input High Level Voltage	All I/O Pins, KBY0-2	0.75xV_{DD}			V
V_{HYS}	Hysteresis Voltage(1)	All I/O Pins, KBY0-2 V_{DD} = 5V		1.0		V
V_{OL}	Low Level Output Voltage	Port B/C, DA0-3, BSW0-3, OUT1, VS, OSD Outputs, V_{DD} = 4.5V I_{OL} = 1.6mA I_{OL} = 5.0mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	Port A V_{DD} = 4.5V I_{OL}= 3.2mA I_{OL}= 30mA			 0.4 1.0	 V V
V_{OL}	Low Level Output Voltage	OSDOSCout, OSCout V_{DD} = 4.5V I_{OL}= 0.1mA			 0.4	 V
V_{OH}	High Level Output Voltage	Port B/C(2), VS V_{DD} = 4.5V I_{OH} = – 1.6mA	 4.1			 V
V_{OH}	High Level Output Voltage	OSDOSCout, OSCout, V_{DD} = 4.5V I_{OL}= – 0.1mA	 4.1			 V
I_{PU}	Input Pull Up Current Input Mode with Pull-up	Port B/C, KBY0-2 V_{IN}= V_{SS}(2)	– 100	– 50	– 25	mA
I_{IL} I_{IH}	Input Leakage Current	OSCin V_{IN}= V_{SS} V_{IN}= V_{DD}	– 10 0.1	– 1 1	– 0.1 10	µA
I_{IL} I_{IH}	Input Leakage Current	All I/O Input Mode no Pull-up OSDOSCin V_{IN}= V_{DD} or V_{SS}	– 10		10	µA
I_{IL} I_{IH}	Input Leakage Current	Reset Pin with Pull-up V_{IN}= V_{SS}	– 50	– 30	– 10	µA
I_{IL} I_{IH}	Input Leakage Current	AFC Pin V_{IH}= V_{DD} V_{IL}= V_{SS} V_{IH}= 12.0V	 – 1		 1 40	µA
I_{OH}	Output Leakage Current	Port A, DA0-3, BSW0-3 OUT1, OSDout V_{OH} = V_{DD}			10	µA
I_{OH}	Output Leakage Current High Voltage	Port A, DA0-3, BSW0-3 OUT1 V_{OH} = 12V			40	µA

DC ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
I_{DD}	Supply Current RUN Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6.0V		6	16	mA
I_{DD}	Supply Current WAIT Mode	f_{OSC}= 8MHz, ILoad= 0mA V_{DD}= 6V		3	10	mA
V_{ON}	Reset Trigger Level ON	RESET Pin			$0.3xV_{DD}$	V
V_{OFF}	Reset Trigger Level OFF	RESET Pin	$0.8xV_{DD}$			V
V_{TA}	Input Level Absolute Tolerance	A/D AFC Pin V_{DD} = 5V			±200	mV
V_{TR}	Input Level Relatice Tolerance	A/D AFC Pin Relative to other levels V_{DD} = 5V			±100	mV

AC ELECTRICAL CHARACTERISTICS

(T_A = 0 to +70°C, f_{OSC}=8MHz, V_{DD}=4.5 to 6.0V unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
t_{WRES}	Minimum Pulse Width	RESET Pin	125			ns
tO_{HL}	High to Low Transition Time	PA6, PA7 V_{DD} = 5V, CL = 1000pF (2)		100		ns
tO_{HL}	High to Low Transition Time	DA0-DA5, PB0-PB6, OSD Outputs, PC0-PC7, V_{DD} = 5V, CL = 100pF		20		ns
tO_{LH}	Low to High Transition Time	PB0-PB6, PA0-PA3, OSD Outputs, PC0-PC3 V_{DD} = 5V, CL = 100pF		20		ns
tO_H	Data HOLD Time SPI after clock goes low I^2CBUS/S-BUS Only		175			ns
f_{DA}	D/A Converter Repetition Frequency(1)			31.25		kHz
f_{SIO}	SIO Baud Rate(1)			62.50		kHz
t_{WEE}	EEPROM Write Time	T_A = 25°C One Byte		5	10	ms
Endurance	EEPROM WRITE/ERASE Cycles	$Q_A L_{OT}$ Acceptance Criteria	300.000	> 1 million		cycles
Retention	EEPROM Data Retention (4)	T_A = 25°C	10			years
C_{IN}	Input Capacitance (3)	All Inputs Pins			10	pF
C_{OUT}	Output Capacitance (3)	All outputs Pins			10	pF
COSCin, COSCout	Oscillator Pins Internal Capacitance(3)			5		pF
COSDin, COSDout	OSD Oscillator External Capacitance	Recommended	15		25	pF

Notes:

1.A clock other than 8 MHz will affect the frequency response of those peripherals (D/A, 62.5kHz and SPI) whose clock is derived from the system clock.

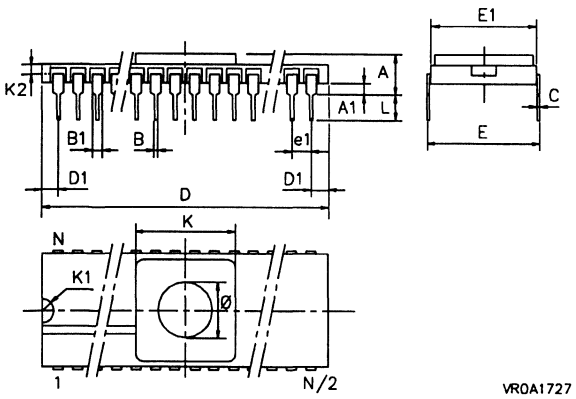
2. The rise and fall times of PORT A have been reduced in order to avoid current spikes while maintaining a high drive capability

3. Not 100% Tested

4. Based on extrapolated data

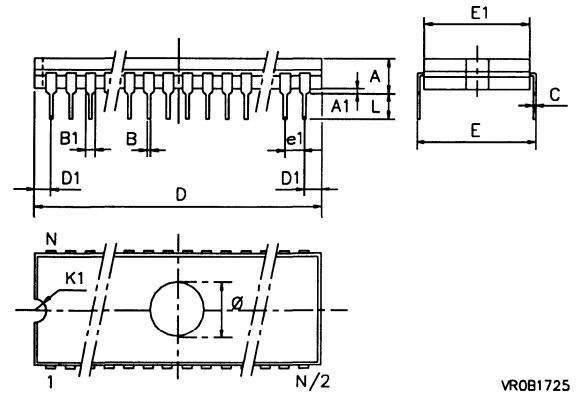
PACKAGE MECHANICAL DATA

Figure 69. 40 Pin Ceramic Dual-In-line Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A						
A1						
B		0.45			.018	
B1						
C						
D		50.8			2.00	
D1						
E		15.2 BSD			0.600 BSD	
E1						
K		13.2			.52	
L						
e1						
Ø						
	Number of Pins					
N	40					

Figure 70. 28-Pin Ceramic Dual-In-line Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.71			.225
A1	0.50		1.78	.020		.070
B	0.40		0.55	.016		.022
B1	1.17		1.42	.046		.056
C	0.22		0.31	.009		.012
D			38.10			1.500
D1	15.2		24.9	.060		.098
E	15.2 BSD			0.600 BSD		
E1	13.05		13.36	.514		.526
L	3.00			.118		
e1	2.29		2.79	.090		.110
Ø	6.86		7.36	.270		.290
	Number of Pins					
N	28					

ORDERING INFORMATION

To ensure compatibility between the EPROM/OTP parts and the corresponding ROM families, the following information is provided. the user should take this information into account when programming the memory and OSD characters of the EPROM parts.

Communication of the ROM Codes. To communicate the contents of memories to SGS-THOMSON, the customer has to send:

- one file in INTEL INTELLEC 8/MDS FORMAT (either as an EPROM or in a MS-DOS 5" diskette) for the ODD and EVEN OSD Character OSD ROM/EEPROM
- one file in INTEL INTELLEC 8/MDS FORMAT (either as an EPROM or in a MS-DOS 5" diskette) for the EEPROM initial content (this file is optional)
- a filled Option List form as described in the OPTION LIST paragraph.

The ROM code must be generated with ST6 assembler. Before programming the EPROM, the buffer of the EPROM programmer must be filled with FFh.

For shipment to SGS-THOMSON the EPROMs should be placed in a conductive IC carrier and packaging carefully.

Customer EEPROM Initial Contents: Format

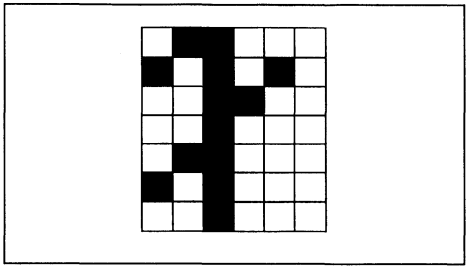
a. The content should be written into an INTEL INTELLEC format file.

b. Undefined or don't care bytes should have the content FFh.

OSD Test Character. IN ORDER TO ALLOW THE TESTING OF THE ON-CHIP OSD MACROCELL THE FOLLOWING CHARACTER MUST BE PROVIDED AT THE FIXED 3Fh (63) POSITION OF THE SECOND OSD BANK.

Listing Generation & Verification. When SGS-THOMSON receives the files, a computer listing is generated from them. This listing refers extractly to the mask that will be used to produce the microcontroller. Then the listing is returned to the customer that must thoroughly check, complete, sign and return it to SGS-THOMSON. The signed list constitutes a part of the contractual agreement for the creation of the customer mask. SGS-THOMSON sales organization will provide detailed information on contractual points.

Figure 71. OSD Test Character



ST63E1xx/T1xx MICROCONTROLLER OPTION LIST

Customer: .....
Address: .....
Contact: .....
Phone No: .....
Reference: .....

Device [ ] ST63E140 [ ] ST63E142 [ ] ST63E126 [ ] ST63E156
[ ] ST63T140 [ ] ST63T142 [ ] ST63T126 [ ] ST63T156

Temperature Range 0 to 70°C

For marking one line with 10 characters maximum is possible
Special Marking [ ] No
[ ] Yes Line1 "_ _ _ _ _ _ _ _ _ _"
Letters, digits, ' . ', ' - ', ' / ' and spaces only
the default marking is equivalent to the sales type only (part number).

CHECK LIST:

	YES	NO
OSD Code: ODD & EVEN	[ ]	[ ]
EEPROM Code (if Desired)	[ ]	[ ]

Signature .....

Date .....

ORDERING INFORMATION TABLE

Sales Type	ROM/EEPROM	Temperature Range	Package
ST63E140D1	8K (EPROM) / 128 Bytes	0 to + 70 °C	CDIP28
ST63E142D1		0 to + 70 °C	CDIP28
ST63E126D1		0 to + 70 °C	CDIP40
ST63E156D1		0 to + 70 °C	CDIP40
ST63T140B1	8K (OTPROM) / 128 Bytes	0 to + 70 °C	PDIP28
ST63T142B1		0 to + 70 °C	PDIP28
ST63T126B1		0 to + 70 °C	PDIP40
ST63T156B1		0 to + 70 °C	PDIP40

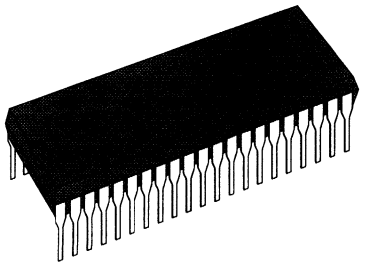
SGS-THOMSON MICROELECTRONICS

ST7271

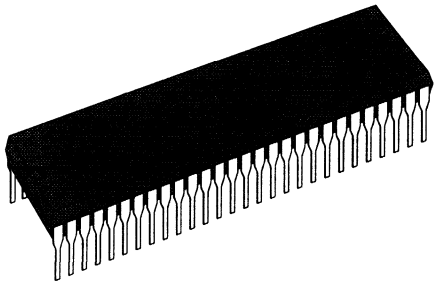
8-BIT HCMOS MCUs WITH EEPROM AND TV/MONITOR DEDICATED FUNCTIONS

PRELIMINARY DATA

- 5V ± 10% supply operating range
- 4MHz Maximum Internal Clock Frequency
- Fully static operation
- 0 to +70°C Operating Temperature Range
- Run, Wait, and Stop Modes
- User ROM: up to 15144 bytes
  Data RAM: up to 256 bytes
  EEPROM: up to 512 bytes
  EWPCC EEPROM: 256 bytes
- 56 pin Shrink Dual In Line Package (ST7271N)
  42 pin Shrink Dual In Line Package (ST7271J)
- up to 27 I/O lines
  8 I/O Open Drain with 12V capability
- up to 8 lines programmable as interrupt wake-up inputs
- 16-bit timer with 2 input capture and 2 output compare functions
- Sync Processor for video timing analysis
- East/West Pin Cushion Automatic Correction with DAC output.
- Watchdog for system reliability and integrity
- 8-bit Analog to Digital Converter with up to 8 channels
- 16 10-bit PWM/BRM Digital to Analog outputs
- 2 12-bit PWM/BRM Digital to Analog outputs
- Industry Standard Serial Peripheral Interface
- User mask options:
  - SPI Data Rate
  - Watchdog enable/disable after Reset
  - Watchdog enable during WAIT mode
- Master Reset and Power-on reset
- Full Hardware Emulator
- 8-bit data manipulation
- 74 basic instructions
- 10 main addressing modes
- 8x8 unsigned multiply instruction
- true bit manipulation
- Complete development support on Real-time emulator with PC/DOS
- Full software package (Cross Assembler, debugger)



PSDIP42



PSDIP56

(Ordering Information at the end of the Datasheet)

DEVICE SUMMARY

DEVICE	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	PACKAGE
ST7271N5	16K	256	512	PSDIP56
ST7271N3	12K	256	512	PSDIP56
ST7271N1	8K	192	384	PSDIP56
ST7271J1	8K	192	384	PSDIP42

Figure 1a. 56 Pin Shrink DIP Pinout

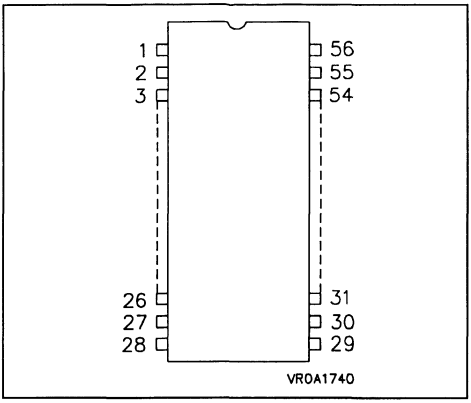
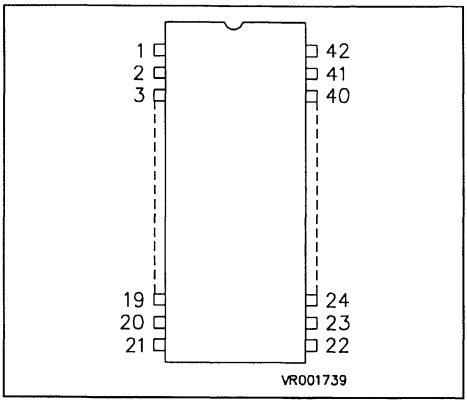


Figure 1b. 42 Pin Shrink DIP Pinout



ST7271N Pin Description

Pin	Name	Pin	Name
1	V_{DDA}	56	V_{SSA}
2	EWPCC	55	V_{SS}
3	DA0	54	PC5/$\overline{SS}$
4	DA1	53	PC4/SCK
5	DA2	52	PC3/MOSI
6	DA3	51	PC2/MISO
7	DA4	50	PC1
8	DA5	49	PC0/OCMP
9	DA6	48	TEST
10	DA7	47	DA17
11	DA8	46	DA16
12	DA9	45	DA15
13	PB7	44	DA14
14	PB6	43	PA0
15	PB5	42	PA1
16	PB4	41	PA2
17	PB3	40	PA3
18	PB2	39	PA4
19	PB1	38	PA5
20	VFBACK/PB0	37	PA6
21	PD4	36	PA7
22	CLMPO/PD3	35	DA13
23	DA10	34	DA12
24	DA11	33	OSCIN
25	$\overline{RESET}$	32	OSCOUT
26	VSYNCO/PD2	31	CSYNCI/PDO
27	VSYNCI	30	HSYNCO/PD1
28	V_{DD}	29	HSYNCI

ST7271J Pin Description

Pin	Name	Pin	Name
1	V_{DDA}	42	V_{SSA}
2	EWPCC	41	V_{SS}
3	DA0	40	PC5/$\overline{SS}$
4	DA1	39	PC4/SCK
5	DA2	38	PC3/MOSI
6	DA3	37	PC2/MISO
7	DA4	36	PC0/OCMP
8	DA5	35	TEST
9	DA6	34	PA0
10	DA7	33	PA1
11	DA8	32	PA2
12	DA9	31	PA3
13	PB3	30	PA4
14	PB2	29	PA5
15	PB1	28	PA6
16	VFBACK/PB0	27	PA7
17	CLMPO/PD3	26	OSCIN
18	$\overline{RESET}$	25	OSCOUT
19	VSYNCO/PD2	24	CSYNCI/PDO
20	VSYNCI	23	HSYNCO/PD1
21	V_{DD}	22	HSYNCI

1 GENERAL DESCRIPTION

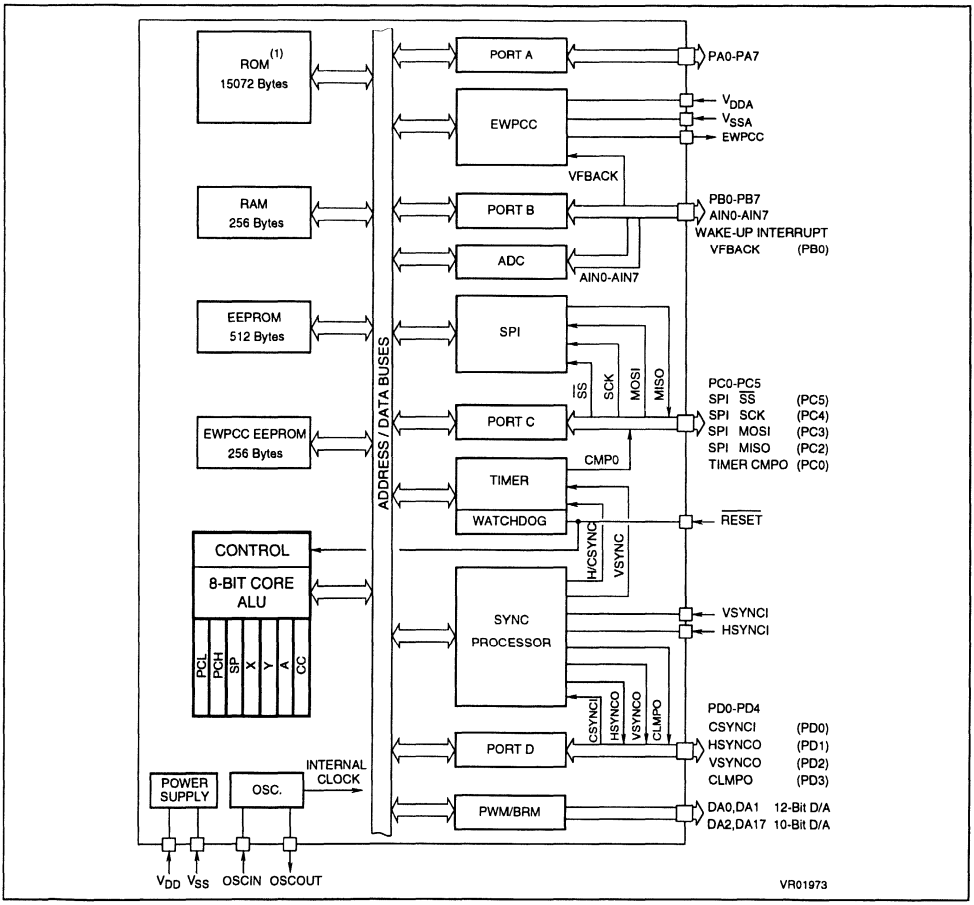
1.1 INTRODUCTION

The ST7271 is a HCMOS microcontroller unit (MCU) from the ST72 family with dedicated peripherals for TV and Monitor applications.
It is based around an industry standard 8-bit core and offers an enhanced instruction set. The processor runs with an external clock at 8 MHz with a 5V supply. Due to the fully static design of this device, operation down to DC is possible. Under software control the ST7271 can be placed in WAIT or STOP mode thus reducing power consumption. The enhanced instruction set and addressing modes afford real programming potential.
In addition to standard 8-bit data management the ST7271 features true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

The device includes an on-chip oscillator, CPU, ROM, RAM, EEPROM, I/O, a timer with 2 input capture and 1 output compare signals, an 8-channel Analog to Digital Converter and an industry standard SPI as standard peripherals.

Dedicated functions include a Sync Processor for video timing analysis, East-West Pin Cushion automatic correction and 18 PWM/BRM outputs for analog control of external functions.

Figure 2. ST7271 Block Diagram



Note 1 : ROM is replaced by EPROM for EPROM/OTP versions.

1.2 PIN DESCRIPTION

V_{DD}. Power supply voltage

V_{SS}. Digital Ground

V_{DDA}. Analog V_{DD} and reference for EWPCC Digital to Analog Converter (DAC, 8 Volts).

V_{SSA}. Analog V_{SS} for EWPCC DAC.

OSCIN, OSCOUT. Oscillator input and output pins. These pins are to be connected to a parallel resonant crystal or ceramic resonator. An external clock source can also be input on OSCIN.

$\overline{\text{RESET}}$. The active low input signal forces the initialization of the MCU. This event is the top priority non maskable interrupt. This pin is switched low when the Watchdog has triggered. It can be used to reset external peripherals.

TEST. This pin must be held low for normal operation

VFBACK (PB0). Vertical Flyback signal (TTL level). This pin accepts the Vertical Flyback signal used for timing correlation for the East-west Pin Cushion correction when this is used or is PB0.

EWPCC. Analog output of correction signal from East-West Pin Cushion controller (2-6V, I_{OUT} = 1mA).

OCMP (PC0). Output compare signal coming from the TIMER. This output signal, according to a register bit option, can be the OCMP pin (for output compare 1 of the timer) or the PC0 pin.

MISO (PC2). SPI Master Out/Slave In Data Output/Input when SPI is enabled or PC2.

MOSI (PC3). SPI Master In/Slave Out Data Input/Output when SPI is enabled or PC3.

SCK (PC4). SPI Serial Clock when SPI is enabled or PC4

$\overline{\text{SS}}$ (PC5). SPI Slave Select when SPI is enabled or PC5.

VSYNCI. Vertical Sychronization Input (TTL level)

HSYNCI. Horizontal Sychronization Input (TTL level)

CSYNCI (PD0). Composite Sychronization Input (TTL level). This pin accepts the composite synchronisation input when the Sync Processor I/O functions are enabled or is PD0.

HSYNCO (PD1). Horizontal Sychronization Output. This pin outputs the horizontal synchronisation output from the Sync Processor (or HSYNCI) when the Sync Processor I/O functions are enabled or is PD1.

VSYNCO (PD2). Vertical Sychronization Output. This pin outputs the vertical synchronisation output from the Sync Processor (or VSYNCI) when the Sync Processor I/O functions are enabled or is PD2.

CLMPO (PD3). Clamp Output. This pin outputs the clamping (back porch) output signal from the Sync Processor (or HSYNCI) when the Sync Processor I/O functions are enabled or is PD3.

DA2-DA17 (56-pin package),
DA2-DA9 (42-pin package), 10-bit PWM/BRM outputs (for Analog controls, after external filtering)

DA0, DA1. 12-bit PWM/BRM outputs (for Analog Controls, after external filtering).

PA0-PA7, PB0-PB7, PC0-PC5, PD0-PD4 (56 pin package). These 27 lines are standard I/O lines, programmable as either input or output.

- **PORT A.** 8 I/O lines, bit programmable, accessed through DDRA and DRA Registers. Each bit can be defined as a standard input port bit without pull-up resistor or as an open drain output port (up to 12V).

- **PORT B**. 8 Standard I/O lines bit programmable accessed through DDRB and DRB Registers. Each bit can be programmed as an analog input (by control bits in the PORT B Configuration register), digital input (with internal pull-up resistor), push-pull digital output or as interrupt wake-up (with pull-up). These negative edge or low-level sensitive interrupt lines can wake-up the ST7271 from WAIT or STOP mode. This feature allows to build low power applications when the ST7271 can be waken-up from keyboard push.
PB0 is used for the East-West Pin cushion controller VFBACK input as shown above when the EWPCC is used.

- **PORT C.** 6 Standard I/O lines accessed through DDRC and DRC Registers. Each bit can be programmed as digital input (with or without pull-up internal resistor), open drain output or SPI control and data signals (as shown for the dedicated SPI signals above). Whenever the SPI is active, the outputs are in the pull-pull configuration.
The pull-up resistor is enabled for all bits present by one control bit in the Programmable Input/Output Configuration Register. The resistor is automatically disabled for the pins used for the SPI when the SPI is enabled.

- **PORT D.** 4 Standard I/O lines bit programmable accessed through DDRD and DRD Registers. Each bit can be programmed as an input (with internal pull-up resistor), push-pull output or Synchronization inputs and outputs to/from the Sync Processor. When programmed as inputs, Video Synchronisation signals can be directly inspected. The inputs may also be passed through the Sync Processor to the Timer Input Captures

These pin functions are also summarised in the following table, which also indicates the availability of functions for the 42-pin SDIP package.

PIN DESCRIPTION (Continued)

Table 1. ST7271 Pin Description

Pin Name	Pin Function(s)	56 Pins	42 Pins
V_{DDA}	Analog V_{DD} for EWPCC	1	1
EWPCC	EWPCC output voltage	2	2
DA0	12-bit PWM/BRM output*	3	3
DA1	12-bit PWM/BRM output	4	4
DA2	10-bit PWM/BRM output*	5	5
DA3	10-bit PWM/BRM output	6	6
DA4	10-bit PWM/BRM output	7	7
DA5	10-bit PWM/BRM output	8	8
DA6	10-bit PWM/BRM output	9	9
DA7	10-bit PWM/BRM output	10	10
DA8	10-bit PWM/BRM output	11	11
DA9	10-bit PWM/BRM output	12	12
PB7	I/O Port PB7	13	
PB6	I/O Port PB6	14	
PB5	I/O Port PB5	15	
PB4	I/O Port PB4	16	
PB3	I/O Port PB3	17	13
PB2	I/O Port PB2	18	14
PB1	I/O Port PB1	19	15
VFBACK/PB0	I/O Port PB0 /VFBACK Input	20	16
PD4	I/O Port PD4	21	
CLMPO/PD3	I/O Port PD3/Clamp Output	22	17
DA10	10-bit PWM/BRM output 10	23	
DA11	10-bit PWM/BRM output 11	24	
$\overline{\text{RESET}}$	Reset Input/Output	25	18
VSYNCO/PD2	I/O Port PD2/VSYNC Output	26	19
VSYNCI	VSYNC Input to Sync Processor	27	20
V_{DD}	Power Supply	28	21
HSYNCI	HSYNC Input to Sync Processor	29	22
HSYNCO/PD1	I/O Port PD1/HSYNC Output	30	23
CSYNCI/PD0	I/O Port PD0/CSYNC Input	31	24
OSCOUT	Oscillator Output	32	25
OSCIN	Oscillator Input	33	26
DA12	10-bit PWM/BRM output 12	34	
DA13	10-bit PWM/BRM output 13	35	
PA7	I/O Port PA7	36	27
PA6	I/O Port PA6	37	28
PA5	I/O Port PA5	38	29
PA4	I/O Port PA4	39	30
PA3	I/O Port PA3	40	31
PA2	I/O Port PA2	41	32
PA1	I/O Port PA1	42	33
PA0	I/O Port PA0	43	34
DA14	10-bit PWM/BRM output 14	44	
DA15	10-bit PWM/BRM output 15	45	
DA16	10-bit PWM/BRM output 16	46	
DA17	10-bit PWM/BRM output 17	47	
TEST	TEST input, must be held to V_{SS}	48	35
PC0/OCMP	I/O Port PC0, Timer Output Compare	49	36
PC1	I/O Port PC1	50	
PC2/MISO	I/O Port PC2, SPI Data	51	37
PC3/MOSI	I/O Port PC3, SPI Data	52	38
PC4/SCK	I/O Port PC4, SPI Clock output	53	39
PC5/$\overline{\text{SS}}$	I/O Port PC5, SPI Slave Select	54	40
V_{SS}	Digital ground	55	41
V_{SSA}	Analog ground for EWPCC	56	42

Note * : Open Drain

1.3 CENTRAL PROCESSING UNIT

1.3.1 Introduction

The CPU has a full 8-bit parallel architecture. Six internal registers allow efficient 8-bit data manipulations. The CPU is able to execute 74 basic instructions with 9 main addressing modes. It is able to address 16k bytes of memory and registers with its program counter.

1.3.2 CPU Registers

The 6 CPU registers are shown in the programming model in Figure 3. Following an interrupt, all registers except Y are pushed onto the stack in the order shown in Figure 4. They are popped from stack in the reverse order.
The Y register is not affected by these automatic procedures. The interrupt routine must therefore handle Y, if needed, through the POP and PUSH instructions.

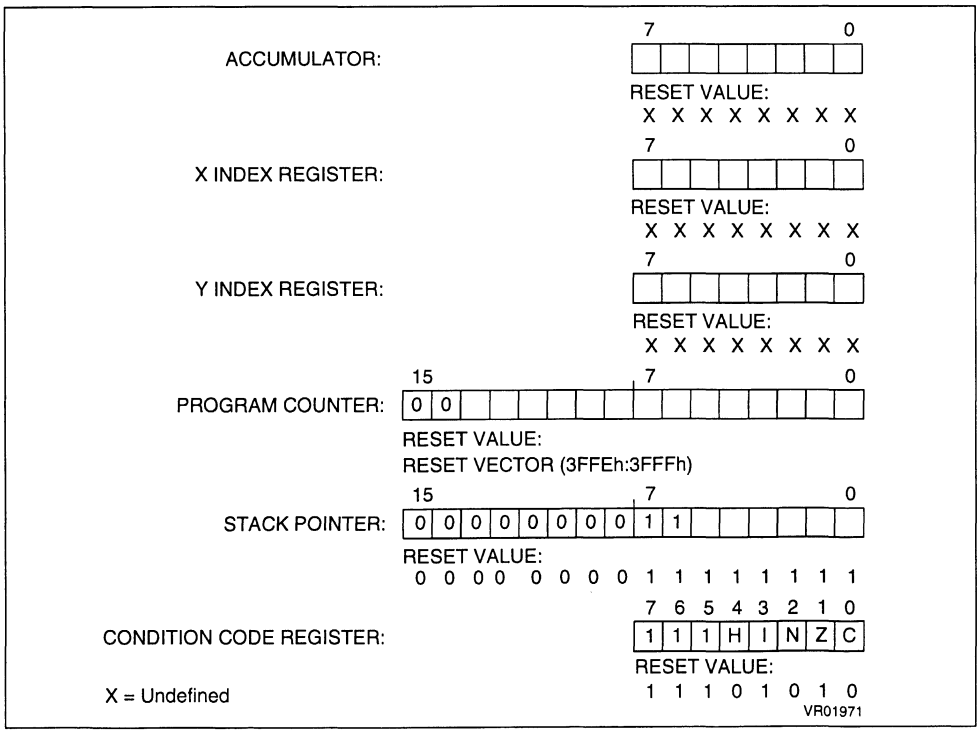
Accumulator (A). The accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations as well as data manipulations.

Index Registers (X and Y). These 8-bit registers are used to create effective addresses or as temporary storage area for data manipulations. The Y register is never automatically stacked. Interrupt routines must push or pop it by using the POP and PUSH instructions.

Program Counter (PC). The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. In the ST7271, only the 14 low order bits are used, bits 14 and 15 are forced to '0' giving an addressing range of 0- 3FFFh.

Stack Pointer (SP). The stack pointer is a 16-bit register. The 6 least significant bits contain the address of the next free location of the stack. The 10 most significant bits are forced as indicated in Figure 3.

Figure 3. Programming Model



CENTRAL PROCESSING UNIT (Continued)

The stack is used to save the CPU context on subroutines calls or interrupts. The user can also directly manipulate the stack through the PUSH and POP instructions.

After a MCU reset or after the reset stack pointer instruction (RSP), the stack pointer is set to its upper value (FFh). It is then decremented after data has been pushed onto the stack and incremented after data is popped from the stack. When the lower limit is exceeded, the stack pointer wraps around to the stack upper limit. The previously stored information is then overwritten and therefore lost.

A subroutine call occupies two stack locations and an interrupt five locations.

1.3.3 Condition Code Register (CC).

The condition code register is a 5 bit register which indicates the result of the instruction just executed as well as the state of the processor. These bits can be individually tested by a program and specified action taken as a result of their state. The following paragraphs describe each bit.

Half carry bit (H). The H bit is set to 1 when a carry occurs between the bits 3 and 4 of the ALU during an ADD, ACC, SUB or SBC instructions. The H bit is useful in BCD arithmetic subroutines.

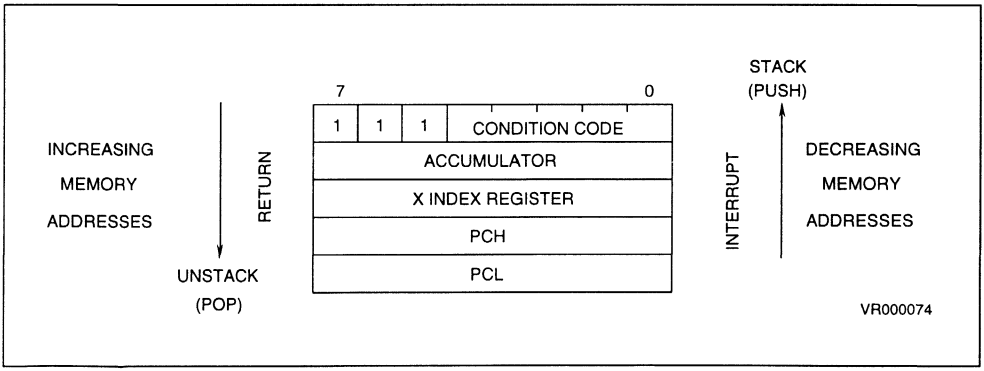
Interrupt mask (I). When the I bit is set to 1, all interrupts except the TRAP software interrupt are disabled. Clearing this bit enables interrupts to be passed to the core. Interrupts requested while I is set are latched and can be processed when I is cleared (only one interrupt request per interrupt enable flag can be latched).

Negative (N). When set to 1, this bit indicates that the result of the last data manipulation is negative (i.e. the most significant bit is a logic 1).

Zero (Z). When set to 1, this bit indicates that the result of the last data manipulation is zero.

Carry/Borrow (C). When set, C indicates that a carry or borrow out of the ALU occurred during the last arithmetic operation. This bit is also affected during bit test and branch, shift and rotate instructions.

Figure 4. Stacking Order



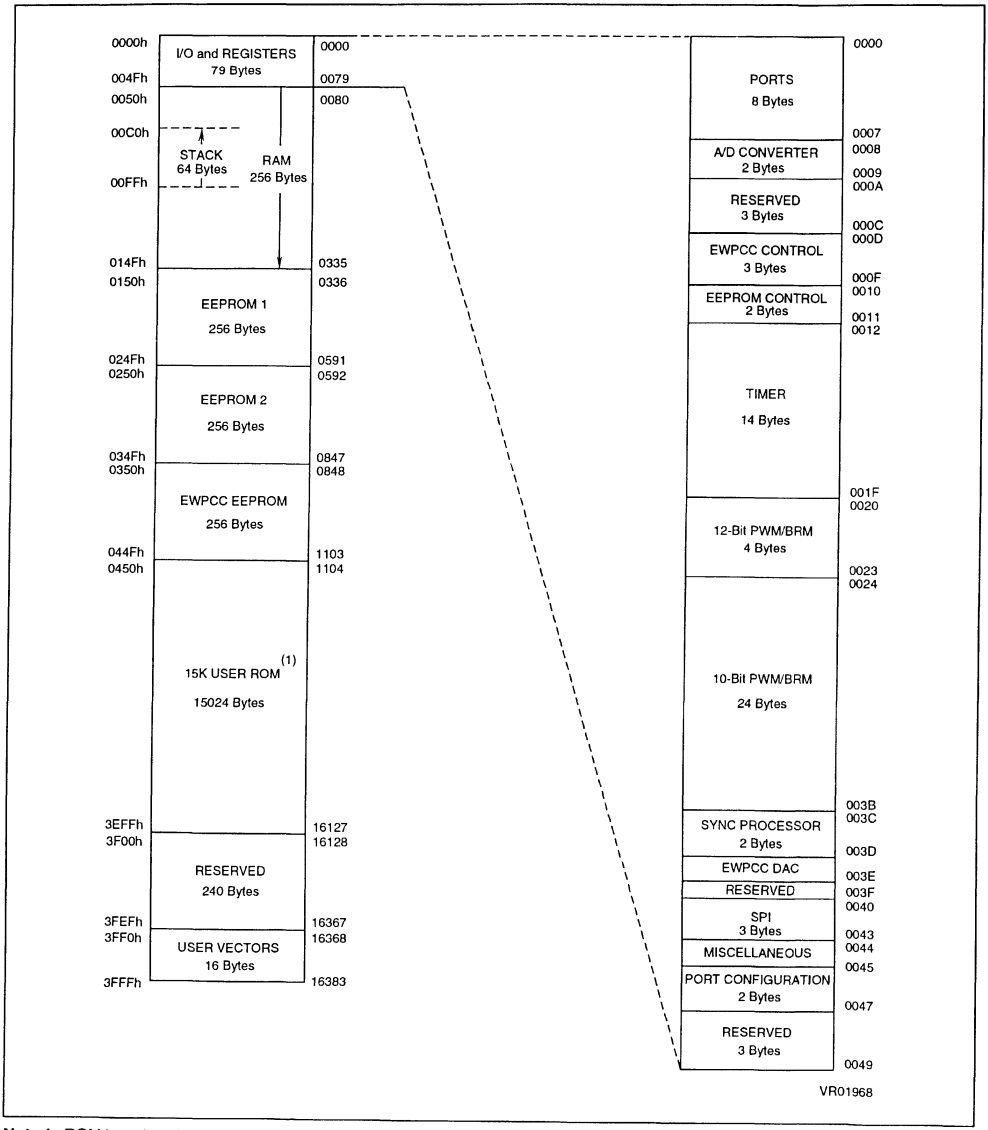
1.4 MEMORY MAP

As shown in Figure 5, the MCU is capable of addressing 16K bytes of memory and I/O registers. In the ST7271, 16383 of these bytes are user accessible.

The available memory locations consist of 80 bytes of I/O registers, 256 bytes of RAM, 512 bytes of EEPROM, 256 bytes of EWPCC EEPROM and 15Kbytes of user ROM. The RAM space includes 64 bytes for the stack from 00FFh to 00C0h.

The highest address bytes contain the user defined reset and interrupt vectors

Figure 5. Memory Map



Note 1 : ROM is replaced by EPROM for EPROM/OTP versions.

1.5 I/O and REGISTER MAP

Address	Register Name
0000h	PORT A DATA REGISTER
0001h	PORT B DATA REGISTER
0002h	PORT C DATA REGISTER
0003h	PORT D DATA REGISTER
0004h	PORT A DATA DIRECTION REGISTER
0005h	PORT B DATA DIRECTION REGISTER
0006h	PORT C DATA DIRECTION REGISTER
0007h	PORT D DATA DIRECTION REGISTER
0008h	A/D DATA REGISTER
0009h	A/D CONTROL/STATUS REGISTER
000Ah-000Ch	Reserved
000Dh	EWPCC0 REGISTER
000Eh	EWPCC1 REGISTER
000Fh	EWPCC EEPROM CONTROL REGISTER
0010h	EEPROM 1 CONTROL REGISTER
0011h	EEPROM 2 CONTROL REGISTER
0012h	TIMER CONTROL REGISTER
0013h	TIMER STATUS REGISTER
0014h	INPUT CAPTURE REGISTER 1, High
0015h	INPUT CAPTURE REGISTER 1, Low
0016h	OUTPUT COMPARE REGISTER 1, High
0017h	OUTPUT COMPARE REGISTER 1, Low
0018h	COUNTER REGISTER, High
0019h	COUNTER REGISTER, Low
001Ah	ALTERNATE COUNTER REGISTER, High
001Bh	ALTERNATE COUNTER REGISTER, Low
001Ch	INPUT CAPTURE REGISTER 2, High
001Dh	INPUT CAPTURE REGISTER 2, Low
001Eh	OUTPUT COMPARE REGISTER 2, High
001Fh	OUTPUT COMPARE REGISTER 2, Low
0020h	PWM0 - 12 bit PWM/BRM
0021h	BRM0 - 12 bit PWM/BRM
0022h	PWM1 - 12 bit PWM/BRM
0023h	BRM1 - 12 bit PWM/BRM
0024h	PWM2 - 10 bit PWM/BRM
0025h	BRM3+BRM2

Address	Register Name
0026h	PWM3
0027h	PWM4
0028h	BRM5+BRM4
0029h	PWM5
002Ah	PWM6
002Bh	BRM7+BRM6
002Ch	PWM7
002Dh	PWM8
002Eh	BRM9+BRM8
002Fh	PWM9
0030h	PWM10
0031h	BRM11+BRM10
0032h	PWM11
0033h	PWM12
0034h	BRM13+BRM12
0035h	PWM13
0036h	PWM14
0037h	BRM15+BRM14
0038h	PWM15
0039h	PWM16
003Ah	BRM17+BRM16
003Bh	PWM17
003Ch	SYNC MUX CONTROL REGISTER
003Dh	SYNC COUNTER CONTROL REGISTER
003Eh	EWPCC DAC REGISTER
003Fh	Reserved
0040h	SPI DATA I/O REGISTER
0041h	Reserved
0042h	SPI CONTROL REGISTER
0043h	SPI STATUS REGISTER
0044h	MISCELLANEOUS REGISTER
0045h	PORT B CONFIGURATION REGISTER
0046h	PROGRAMMABLE INPUT/OUTPUT CONFIGURATION REGISTER
0047h	Reserved
0048h	Reserved
0049h	Reserved

1.6 WATCHDOG SYSTEM

The watchdog system consists in a divider-by-8 counter and a fixed divide-by-1024 prescaler. It is controlled through bit WDOG of the Miscellaneous Register.

Two mask options are provided:

- The WATCHDOG ENABLE mode mask option
- The WATCHDOG DURING WAIT mask option

The Watchdog Enable Mode mask option selects the state of the watchdog system after an external or a power-on reset. In the "programmable enable" option, a reset causes the watchdog to be disabled and the counter to be forced to zero. In the "auto enable" option, the watchdog is automatically enabled after the start-up procedure.

When the watchdog is configured with the "programmable enable" option, the watchdog system is enabled by setting the WDOG bit of the Miscellaneous Register (0044h). Only an external or a power-on reset can clear WDOG and disable the watchdog system.

Whatever the option, when the watchdog counter is enabled, it is driven by the CPU clock through the divide-by-1024 prescaler (i.e. the counter clock period is 1024 CPU clock cycles). It is reset to zero by writing WDOG at 1. A system reset is generated if the counter reaches its maximum count (8). To avoid a system reset, the software must therefore regularly reset the counter at least before the watchdog time from the last clear or from the time the watchdog system has been enabled.

Care has to be taken when enabling the counter ("programmable enable" option only). The prescaler is in an unknown state at the time WDOG is set. The first rising edge can thus be sent to the watchdog counter after a time comprised between 0 and 1024 CPU clock cycles. In this mode, the first reset of the watchdog counter should therefore not occur later than 6x1024 to 7x1024 CPU clock cycles after it has been enabled.

The system reset is generated by pulling down the $\overline{\text{RESET}}$ pin for at least one and a half CPU clock cycle. The state of the $\overline{\text{RESET}}$ pin is re-entered to the reset logic, thus causing an external reset to be issued.

The Watchdog During Wait mask option determines the watchdog function during the WAIT low power mode. In the "active during WAIT" option, the watchdog is kept active, thus able to reset the MCU if it remains in WAIT mode longer than the watchdog timeout period. In the "suspended during WAIT" option, it suspends operation during the WAIT mode and resets its counter. It will then resume operation when exiting the WAIT mode.

The STOP mode is inhibited when the watchdog system is enabled. However if a STOP instruction is executed while it is enabled, a watchdog reset is immediately generated.

MISCELLANEOUS REGISTER (0044h)

Read/Write

Reset Value: 1111 1010 (FAh)

7							0
1	1	1	1	1	VSYNCI	INT	WDOG

b7-b3 = **Unused**, read "1" when accessed.

b2 = **VSYNCI:** *Internal Vsync* (Read-Only) This bit shows the state of the Vsync output by the Sync Processor.

b1 = **INT:** *Interrupt Request.* This bit sets the interrupt configuration for the PORT B wakeup Interrupt Request:

INT = 0 : selects the falling edge option only

INT = 1 : selects the falling edge or low-level option.

WARNING. *This bit can only be written ONCE after reset. Writing to INT is disabled after the first write to the Miscellaneous Register. Bit manipulation instructions should be used with extreme caution when writing to this register.*

b0 = **WDOG:** *Watchdog System.* Whatever the WATCHDOG ENABLE MODE mask option, the watchdog counter is reset when WDOG is written at 1. When the MCU is configured with the "programmable enable" option, the WDOG bit is low after a reset. It must be set to enable the watchdog system. Writing a '0' clears the WDOG bit, but does not change the watchdog condition. Only a reset can clear WDOG.

1.7 SYSTEM CLOCK

General description.

The MCU accepts either a Crystal/Ceramic resonator or an external clock to provide the internal oscillator.
The internal clock (f_{INT}) is derived by a divide-by-2 from the internal oscillator frequency (f_{OSC}).

Crystal. The internal oscillator is designed to interface with an AT cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC}. The circuit shown on Figure 6 is recommended when using a crystal. Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time.

Ceramic Resonator. A ceramic resonator may be used in place of the crystal in low cost applications. The circuit on Figure 6 is recommended when using a ceramic resonator. The manufacturer of the particular ceramic resonator being considered should be consulted for specific information.

External Clock. An external clock should be applied to the OSCIN input with the OSCOUT pin not connected as shown figure 7. The Crystal clock specifications do not apply when using an external clock input. The equivalent specification of the external clock source should be used.

Figure 6. Crystal/Ceramic Resonator

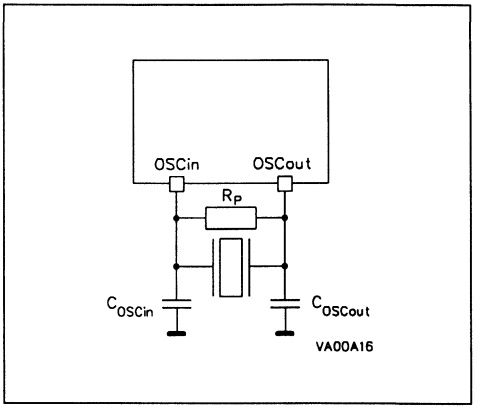


Figure 7. External Clock Source Connections

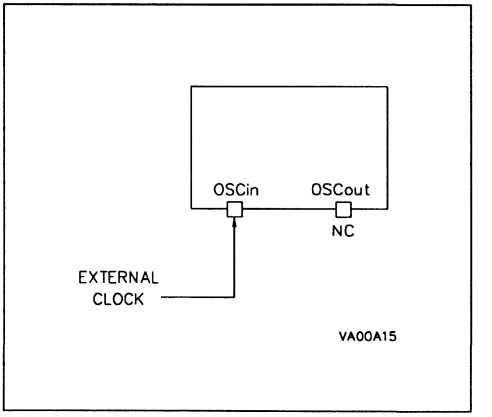
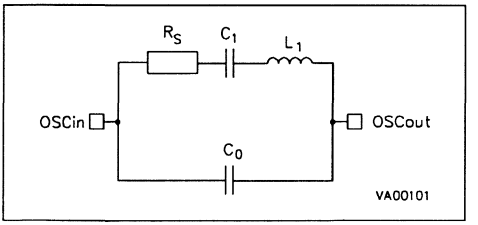


Figure 8. Equivalent Crystal Circuit



Recommended Settings for Crystal

	2MHz	4MHz	Unit
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	pF
C_{OSCin}	15-40	15-30	pF
C_{OSCout}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	10^3

Recommended Settings for Ceramic Resonator

	2-4MHz	Unit
R_{SMAX}	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSCin}	30	pF
C_{OSCout}	30	pF
R_P	1-10	MΩ
Q	1250	

1.8 RESET AND INTERRUPTS

The Reset procedure is used to provide an orderly software startup or to quit low power modes.

Two reset modes are provided: a power-on reset and an external reset at the RESET pin. The watchdog reset is considered as an external interrupt as the watchdog system generates the MCU reset by pulling down the RESET pin.

1.8.1 Power-on Reset (POR)

The power-on reset is generated upon detection of a positive transition on V_{DD} (refer to Figure 9). It causes the reset vector to be fetched from addresses 3FFEh and 3FFFh in order to be loaded into the PC and with program execution starting from this point.

An internal circuitry provides a 4096 CPU clock cycle delay from the time that the oscillator becomes active. At the end of the power-on reset, the MCU can be maintained in the reset condition by holding the external reset low. The RESET pin can therefore be used to ensure V_{DD} has risen to a point where the MCU can properly operate before running the MCU program.

During the POR, the RESET pin is pulled low, thus permitting the MCU to reset other devices.

The power-on reset is strictly used for power up conditions and should not be used to detect any drop in the power supply voltage. There is no internal provision for a power-down reset.

1.8.2 External Reset

The external reset is an active low input signal applied to the RESET pin of the MCU.

As shown in Figure 9, the RESET signal must stay low for a minimum of one and a half CPU clock cycles. A reset causes the reset vector to be fetched at addresses 3FFEh and 3FFFh in order to be loaded into the PC and with program execution starting from this point.

The external reset is also used by the watchdog system to reset the MCU. When active, the power-on reset circuitry pulls down the RESET pin. In both cases, the RESET pin can be used as an output to reset other devices. However, the pull-down circuitry includes current limitation to allow the connection of any input signal, including from an RC type circuit.

An internal Schmitt trigger at the RESET pin is provided to improve immunity to noise.

Table 2. List of sections affected by RESET,WAIT and STOP

Section	RESET	POR	WAIT	STOP
Timer Prescaler reset to zero	X	X	-	-
Timer Counter set to FFFCh	X	X	-	-
All Timer enable bit set to 0 (disable)	X	X	-	-
Data Direction Registers set to 0 (as Inputs)	X	X	-	-
Set Stack Pointer to 00FFh	X	X	-	-
Force Internal Address Bus to restart vector 3FFEh, 3FFFh	X	X	-	-
Set Interrupt Mask Bit (I-Bit, CCR) to 1 (Interrupt Disable)	X	X	-	-
Set Interrupt Mask Bit (I-Bit, CCR) to 0 (Interrupt Enable)	-	-	X	X
Reset STOP Latch	X	X	-	-
Reset INT Latch	X	X	-	-
Reset WAIT Latch	X	X	-	-
Disable Oscillator (for 4096 cycles)	-	X	-	X
Set Timer Clock to 0	-	X	-	X
Watchdog counter reset	X	X	-	X
Watchdog WDOG-BIT reset	X	X	-	X
EEPROM control bits reset	X	X	-	-
PWM/BRM registers reset	X	X	-	-
EWPCC DAC register reset	X	X	-	-
SYNC registers reset	X	X	-	-

RESET AND INTERRUPTS (Continued)

1.8.3 Interrupts

The ST7271 may be interrupted by one of three different methods: three maskable hardware interrupts (PORT B, SPI or TIMER) and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 10.

The maskable interrupts must be enabled in order to be serviced. However, disabled interrupts can be latched and processed when they are enabled. When an interrupt has to be serviced, the PC, X, A and CC registers are saved onto the stack and the interrupt mask (I bit of the Condition Code Register) is set to prevent additional interrupts. The Y register is not automatically saved.

The PC is then loaded with the interrupt vector of the interrupt to service and the interrupt service routine runs (refer to Table 3 for vector addresses). The interrupt service routine should finish with the IRET instruction which causes the contents of the registers to be recovered from the stack and normal processing to resume. Note that the I bit is then cleared if and only if the corresponding bit stored in the stack is zero.

Though many interrupts can be simultaneously pending, a priority order is defined (see Table 3). The RESET pin has the highest priority.

If the I bit is set, TRAP is the only enabled interrupt.

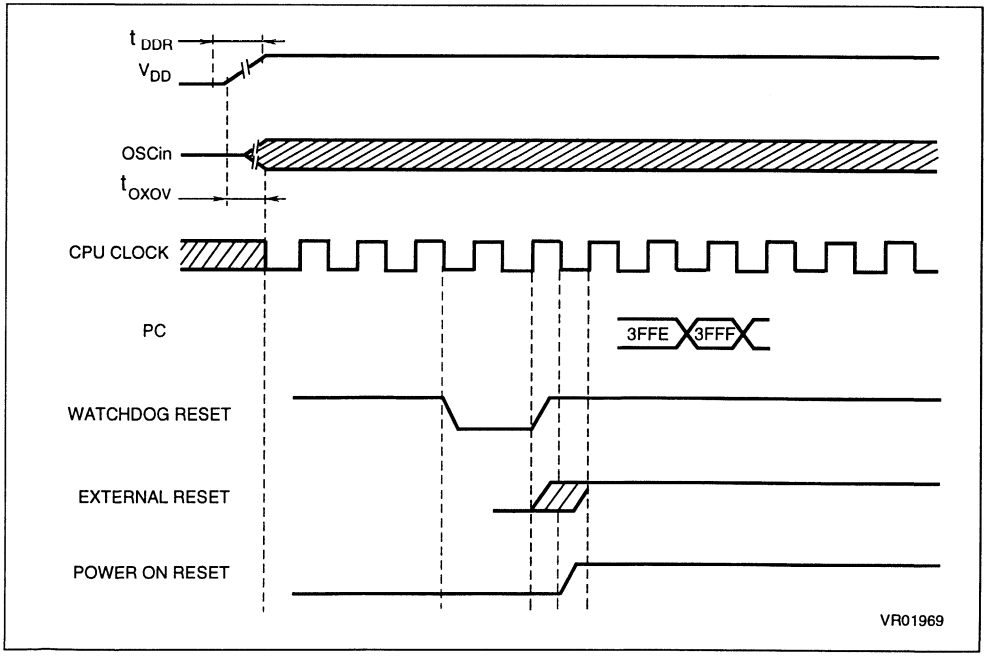
Interrupts allow the processor to leave the Wait low power mode.

Software Interrupt. The software interrupt is the executable instruction TRAP. The interrupt is recognized when the TRAP instruction is executed, regardless of the state of the I bit. When the interrupt is recognized, it is serviced according to the flowchart on Figure 10.

PORTB Interrupt. The PORTB Interrupt can be generated on the falling edge of any pin of PB0-PB7 if it is defined as an interrupt source. When an enabled interrupt occurs, normal processing is suspended at the end of the current instruction execution. It is then processed according to the flowchart on Figure 10.

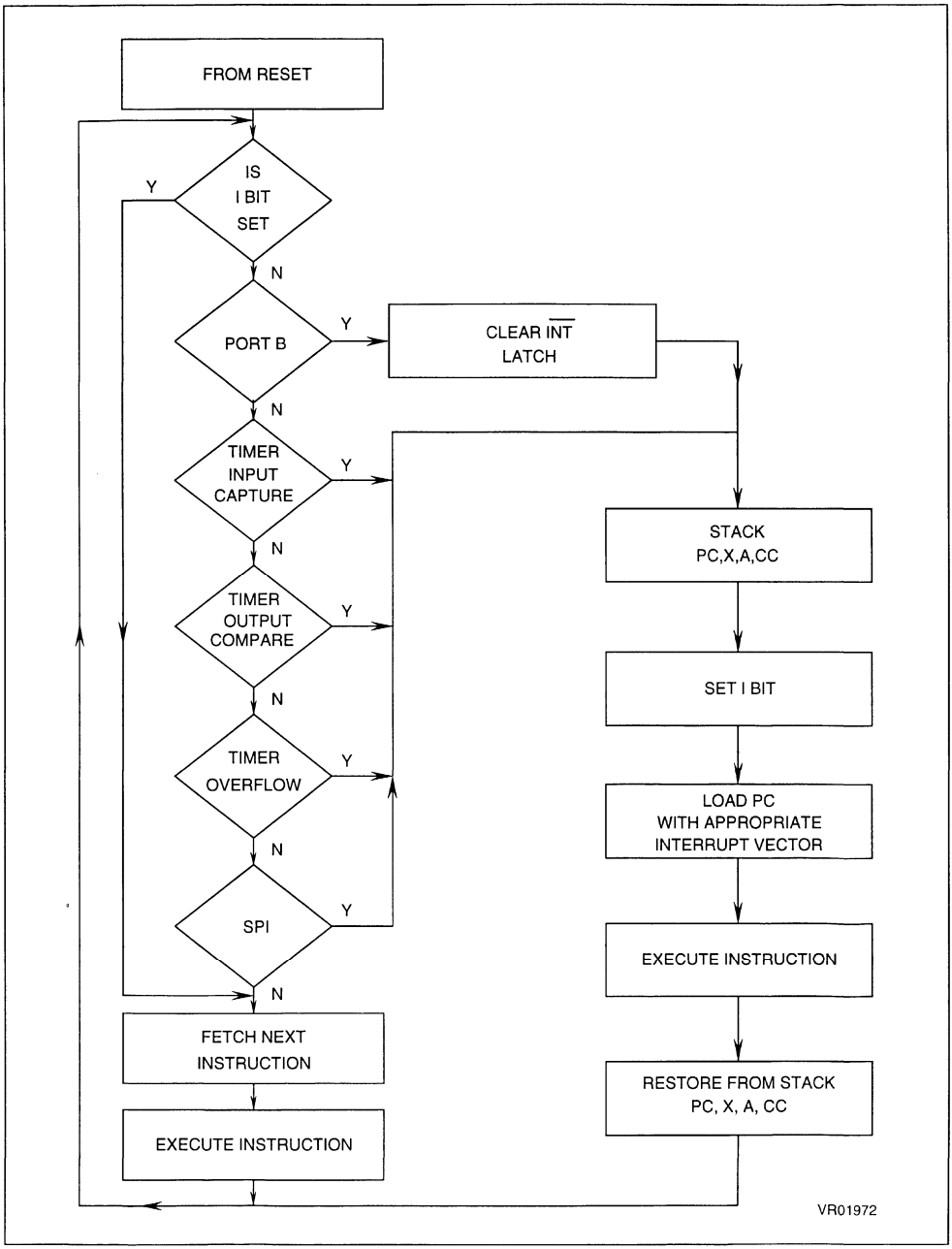
If the interrupt is disabled (I high), the triggering edge of the wake-up interrupt sources logical-ORed is internally latched and the interrupt

Figure 9. Reset Timing Diagram



RESET AND INTERRUPTS (Continued)

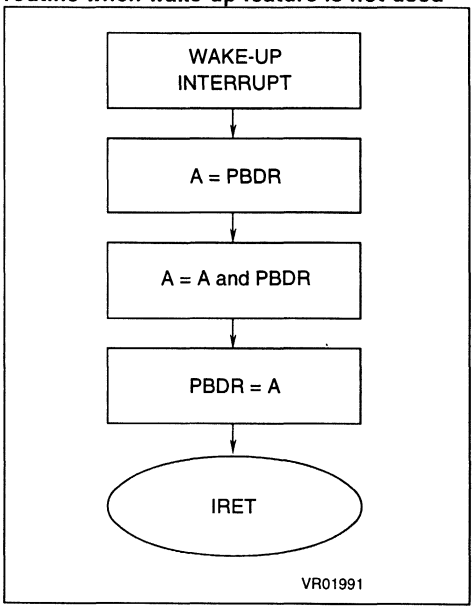
Figure 10. Interrupt Processing Flow-Chart



RESET AND INTERRUPTS (Continued)

remains pending to be processed as soon as the interrupt is enabled. This internal latch is cleared in the first part of the service routine. Therefore, only one external interrupt can be latched and serviced as soon as possible. When the PORTB wake-up function is enabled, care should be taken after a Reset condition interrupt. The flowchart proposed in Figure 11 should be used in these cases.

Figure 11. Proposed Wake-up Interrupt routine when wake-up feature is not used



Timer Interrupt. Five different timer interrupt flags are able to cause a timer interrupt when they are active if both the I bit of the CCR is reset and if the corresponding enable bit is set. If either of these conditions is false, the interrupt is latched and thus remains pending.

The interrupt flags are located in the Timer Status Register (0013h). The Enable bit are in the Timer Control Register (0012h).

When an enabled interrupt occurs, normal processing is suspended at the end of the current instruction execution. It is then serviced according to the flowchart on Figure 10. Software in the timer service routine must determine the priority and cause of the timer interrupt by examining the interrupt flags and the status bits located in the TSR.

The general sequence for clearing an interrupt is an access to the status register while the flag is set followed by a read or write of an associated register. Note that the clearing sequence resets the internal latch. A pending interrupt (i.e. waiting for being enabled) will therefore be lost if the clear sequence is executed.

Serial Peripheral Interface (SPI) Interrupts

An interrupt in the serial peripheral interface (SPI) occurs when one of the interrupt flag bits in the serial peripheral status register (location 0043h) is set, provided the I bit in the condition code register cleared and the enable bit in the serial peripheral control register (location 0042h) is enabled.

When the interrrupt is recognized, the current state of the machine is pushed into the stack and I bit in the condition code register is set. These masks further interrupt until the present one is serviced. The SPI interrupt causes the program counter to vector to memory location 3FF2 and 3FF3 which contains the starting address of the interrupt service routine.

Table 3. Interrupt and Reset priorities

Vector Address	Interrupt Source	Masked by	Priority
3FFEh,3FFFh	RESET and POWER-ON (POR)	none	Highest
3FFCh,3FFDh	SOFTWARE Interrupt (TRAP)	none	
3FFAh,3FFBh	PORT B Wake up	I-Bit	
3FF8h,3FF9h	TIMER INPUT Capture (1 and 2)	I-Bit	
3FF6h,3FF7h	TIMER OUTPUT Compares (1 and 2)	I-Bit	
3FF4h,3FF5h	TIMER OVERFLOW	I-Bit	
3FF2h,3FF3h	SPI	I-Bit	
3FF0h,3FF1h	Reserved	-	Lowest

1.9 LOW POWER MODES

STOP Mode. The STOP mode is the MCU lowest power consumption mode. The STOP mode is entered by executing the STOP instruction. The internal oscillator is then turned off, causing all internal processing to be stopped, including the operation of the on-chip peripherals. The STOP mode cannot be used when the watchdog is enabled, if the STOP instruction is executed while the watchdog system is enabled, a watchdog reset is generated thus resetting the entire MCU.

When entering the STOP mode, the I bit in the Condition Code Register cleared. Thus, the external interrupts are allowed and the MCU is placed at its nominal speed (see CLOCK SYSTEM). All other registers and memory remain unaltered and all I/O lines remain unchanged.

The MCU can exit the STOP mode upon reception of either an external interrupt on PORTB or a power-on or external reset. The oscillator is then turned on and a stabilization time is provided before releasing CPU operation. The stabilization time is 4096 CPU clock cycles.

After the start up delay, the CPU continues operation by servicing the interrupt which wakes it up or by fetching the reset vector if a reset wakes it up.

WAIT Mode. This mode is a low power consumption mode, but the power consumption is higher than in the STOP mode. The consumption can be further reduced by entering the slow mode.

The WFI instruction places the MCU in the WAIT mode.

In the WAIT mode, the internal clock remains active but all CPU processing is stopped; however, all other peripherals are still running. The watchdog can either be active or not according to the WATCHDOG DURING WAIT mask option.

During the WAIT mode, the I bit in the condition code register is cleared to enable all interrupts.

All other registers and memory remain unaltered and all parallel I/O lines remain unchanged.

An interrupt or a reset causes the MCU to exit the WAIT mode. An interrupt while the MCU is in the WAIT mode causes the corresponding interrupt vector to be fetched, the interrupt routine to be executed and normal processing to resume. A reset causes the program counter to fetch the reset vector and processing starts as for a normal reset.

Table 2 gives a list of the different sections affected by the low power modes. For detailed information on a particular devices, please refer to the corresponding parts.

LOW POWER MODES (Continued)

Figure 12a. STOP Flow Chart

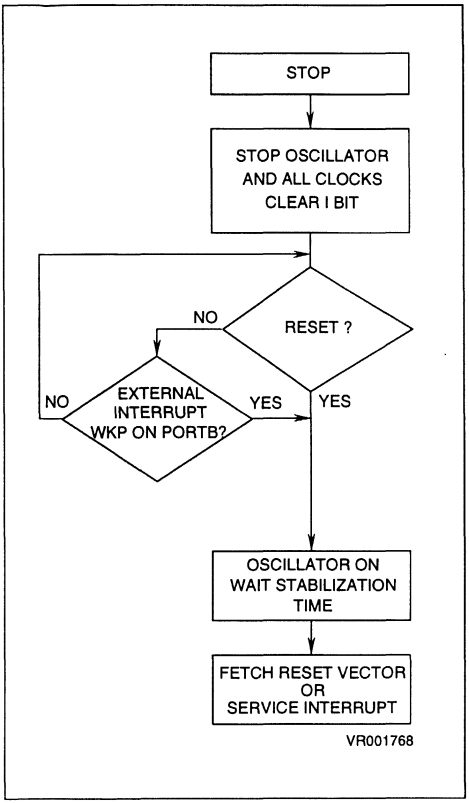
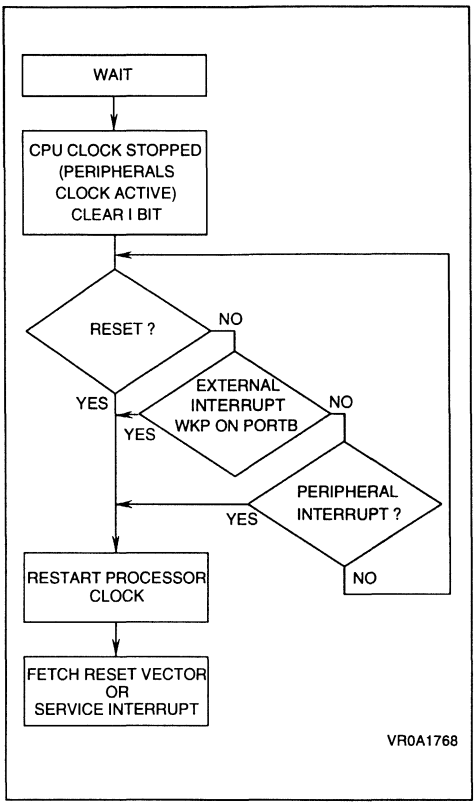


Figure 12b. WAIT Flow Chart



NOTES

2 FUNCTIONS DESCRIPTION

2.1 EEPROM

2.1.1 Introduction

The on-chip EEPROM provides a non-volatile storage of user programmed data. It is read as a normal Read-only memory location, however user programs must not run from the EEPROM.

Programming and erasure are made in conjuction with the EEPROM control registers while 8 data latches allow simultaneous write or erase of 1 to 8 bytes in the EEPROM memory in the same programming cycle. The constraint is that all addressed memory bytes must be on the same row of the EEPROM memory array, that is up to eight bytes with the address bits A7, A6, A5, A4 and A3 constant, and with A2, A1 and A0 selecting the address(es) to be written within the row.

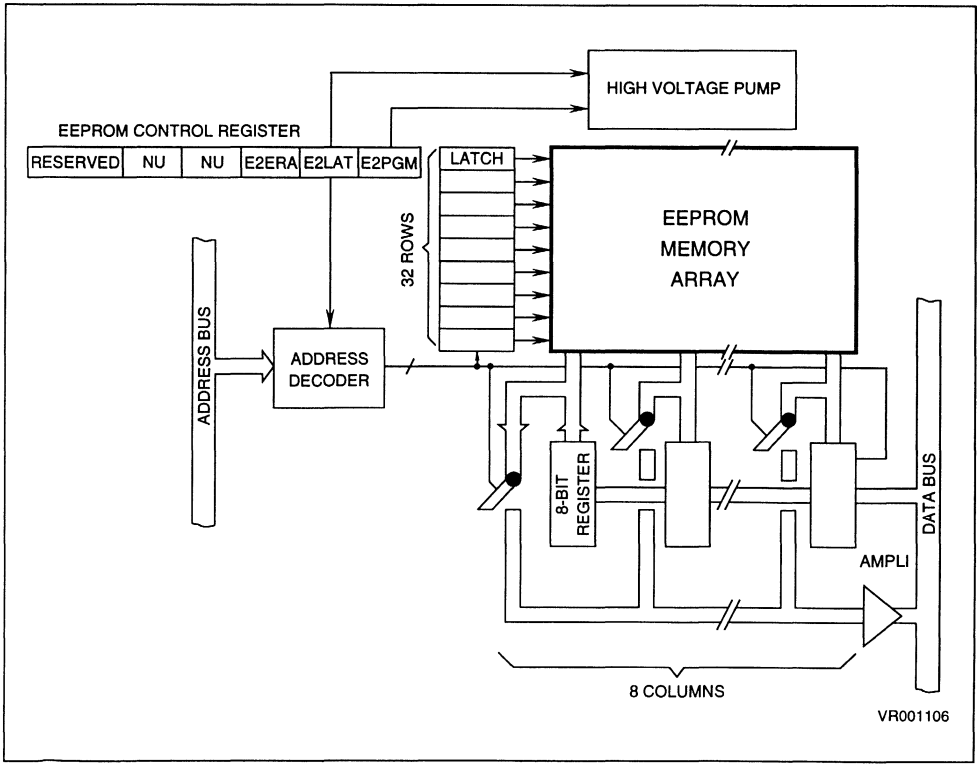
The EEPROM cell includes an internal charge pump to avoid the need of an external high voltage supply for the erase and programming functions.

2.1.2 Functional Description

As shown in Figure 13, the EEPROM is a 8 columns by 32 rows array. The row is selected by the A7, A6, A5, A4, A3 bits. Each column is associated to an 8-bit data register.

Read Operation (E2LAT="0").
The EEPROM can be read as a normal ROM location when the E2LAT bit of the Control Register is low. When E2LAT is low, the E2PGM and E2ERA bits are forced low.

Figure 12. EEPROM Block Diagram



EEPROM (Continued)

Write/Erase Operation. (E2LAT="1")

When E2LAT is set to "1", a write to an EEPROM location latches the data written in the 8-bit register corresponding to the decoded column and sets an internal flag for the decoded row.
As there are 8 columns in each row, up to 8 locations (having the same A7, A6, A5, A4, A3 address bits) can be simultaneously written or erased.

To **erase** bytes, the E2LAT and E2ERA bits are set to "1", and an instruction is made to write to the EEPROM addresses to be erased (the data value is not significant). The E2PGM bit must have been set after this operation to turn the charge pump on.

To **write** bytes, the E2LAT bit is set to "1", and the data is written to the appropriate EEPROM address(es). The E2PGM bit must have been set after this operation to turn the charge pump on.

WARNING a minimum delay of 20µs must be maintained after a programming operation (the falling edge of E2LAT) before the next read or write of the EEPROM. This time is required to discharge the high voltage in the array.

Notes

- Each block of 256 bytes of EEPROM is controlled by an independant EEPROM Control Register. Please refer to the Memory Map for both EEPROM memory block locations.
- It is mandatory to erase bytes before writing them.
- E2LAT must be kept high for the programming time tprog and then be cleared.
- When E2LAT is high, access to the EEPROM array is not possible.
- It is not allowed to perform successive write or erase cycles without clearing E2LAT between each write/erase (see Warning above).

2.1.3 Register Description

EEPROM CONTROL REGISTER 1 (0010h)

EEPROM CONTROL REGISTER 2 (0011h)

Read/Write
Reset Value: 0000 0000 (00h)

7							0
Res	Res	Res	—	—	E2ERA	E2LAT	E2PGM

This register contains the bits required to read, erase and program the EEPROM. They are defined as follow:

b7-5 = **Reserved**, must be held to "0"

b4,3 = **Unused**

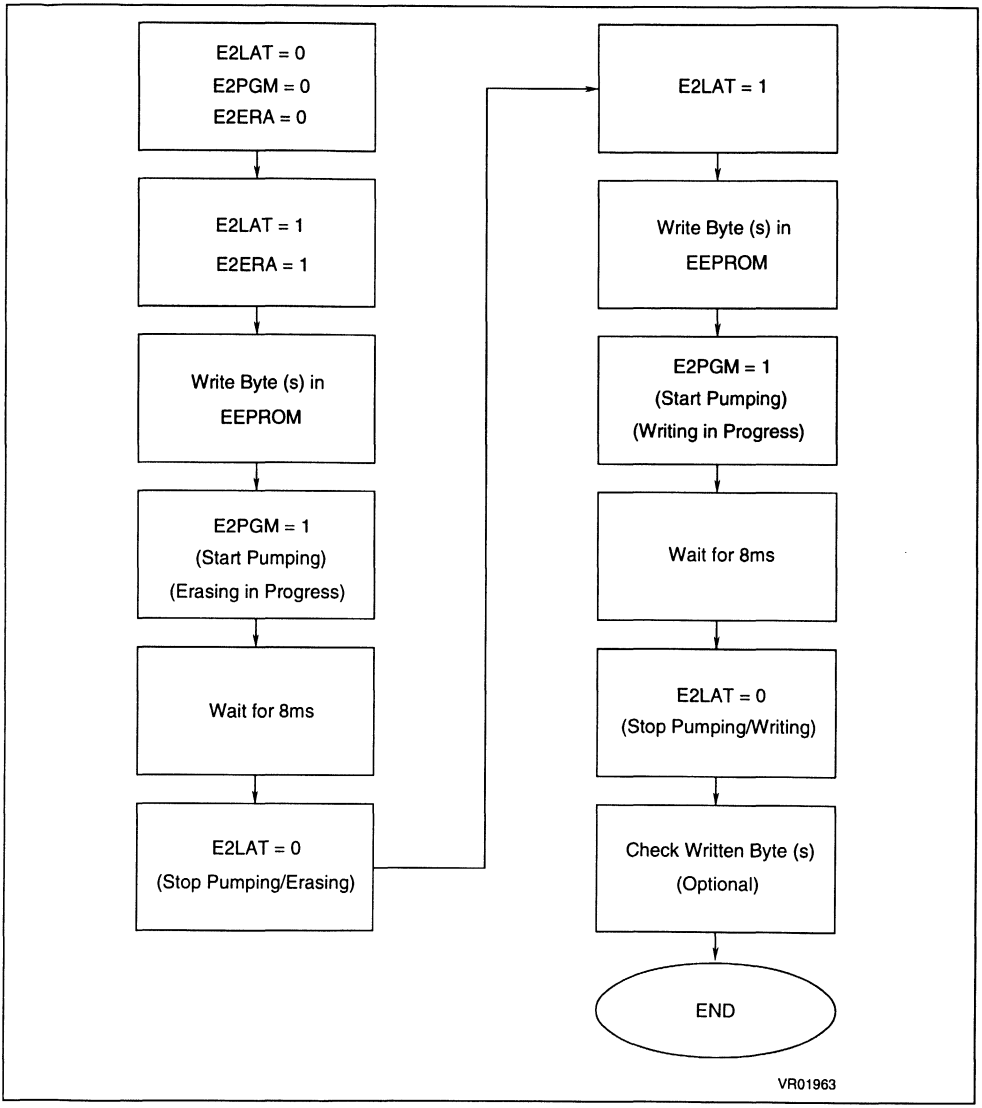
b2 = **E2ERA:** *EEPROM Erase.* E2ERA must be set to "1" for an erase operation. It must be set after or at the same time as E2LAT. It cannot be changed once an EEPROM address is selected. It is held low when E2LAT is low. It is therefore automatically reset when E2LAT is reset.

b1 = **E2LAT:** *EEPROM Latch Enable.* When E2LAT is reset to "0", data can be read from the EEPROM. When it is set to "1" and E2PGM reset to "0", a write into the EEPROM array causes the data to be latched, according to the address into one of 8 data registers. An additional internal flag is latched to select the row. The selected columns and row determine the locations involved in the next erase or programming operation. E2LAT must be cleared after each programming or erase operation. E2ERA and E2PGM are forced low when E2LAT is low.

b0 = **E2PGM:** *EEPROM Program Mode.*This bit allows the internal charge pump to be switched on or off. When set to "1", the charge pump generator is on and the high voltage is applied to the EEPROM array. When low, the charge pump generator is off. E2PGM can only be reset by resetting E2LAT.

EEPROM (Continued)

Figure 13. Programming Flow-chart of a basic routine



2.2 I/O PORTS

2.2.1 Introduction

The I/O ports allow the transfer of data through digital inputs and outputs, and, for specific pins, the input of analog signals or the Input/Output of dedicated signals for the on-chip peripherals (e.g. SPI, EWPCC and Timer). Please refer to the following table for a summary of these Alternate functions.

2.2.2 Functional Description

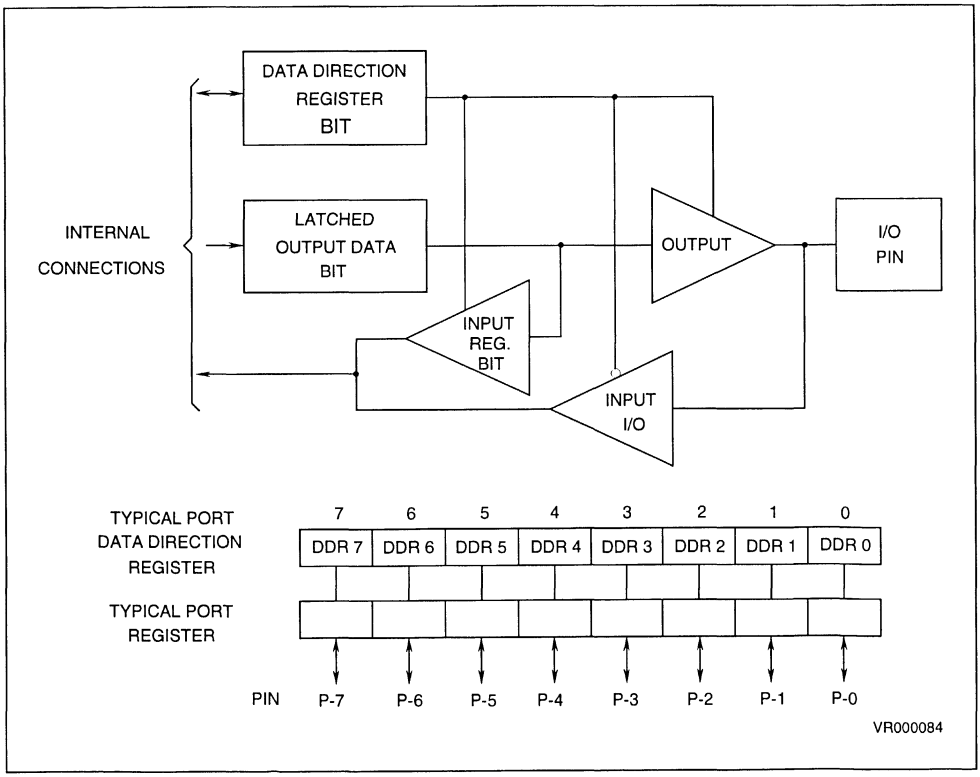
Each port pin of the I/O Ports can be individually configured under software control as either input or output. Ports A, B are 8-bit I/O ports, Port C is a 6-bit I/O port and Port D is a 5-bit port.

Each bit of a Data Direction Register (DDR) corresponds to an I/O pin of the associated port. This corresponding bit must be set to configure its associated pin as output and must be cleared to configure its associated pin as input. The Data Direction Registers can be written or read.

The typical I/O circuit is shown on Figure 15. Any write to an I/O port updates the port output register even if it is configured as input. Any read of an I/O port returns either the data latched in the port output register (pins configured as output) or the value at the I/O pin (pins configured as input) (see Table 6).

At power-on or external reset, all DDR registers are cleared, which configures all Ports B, C and D pins as inputs with pull-ups and Port A as inputs without pull-ups but the port output registers are not initialized. Thus, the I/O port should be written before setting the DDR bit to avoid undefined output levels.

Figure 14. I/O Pin Typical Circuit



I/O PORTS (Continued)

PORT A

Each Port A bit can be defined as an Input line (no pull-up) or as an Output Open drain line capable of handling typical current Isink of 10 mA for LED driving.

PORT B

Each bit of PORT B bit can be used as the Analog source to the Analog to Digital converter by selecting each individual bit independently in the Port B Configuration Register (address 0045h).
When the Analog function is selected for an I/O pin, the pull-up of the respective pin of Port B is disconnected and both the Data and Direction (DDR) registers of the respective pin are reset. Any further accesses to the respective DDR bit is blocked until the pin status is returned to normal I/O.

PORT B bit can also be configured on a bit basis as a wake-up interrupt input with an internal pull-up resistor. This mode is enabled by setting the corresponding Port B bit as a DIGITAL input (its bit in DDR set to '0' and its Analog function disabled) and the corresponding bit in the Port B Data Register must be set to '1'.
When this bit is subsequently forced low, an interrupt will be generated according to the status of the INT bit in the Miscellaneous Register.

Port B, bit 0 is only available for output if the East-West Pin-Cushion controller (EWPCC) is not used. If the EWPCC function is selected, Port B bit 0 MUST be set as input to enable the VFBACK timing input.

The pins not available for the 42 pin package (but present for the 56 pin package) are internally connected as standard digital inputs with pull-ups enabled.

All unused I/O lines should be tied to an appropriate logic level (either V_{DD} or V_{SS}).

PORT C

The available port pins of Port C may be used as general purpose I/O or as the I/O pins of the on-chip SPI and Timer Output Compare.
When used as digital Input, pull-up resistors may be switched on for ALL Port C inputs by setting the PUPC bit of the Programmable Input/Output Configuration Register (PCR).
When used as output the Open Drain mode is automatically set if the SPI is disabled.

Port C, bit 0 is switched from the normal I/O functionality to the output of the Timer Output Compare signal by resetting to '0' the OCOP bit of the PCR.

When the SPI is enabled, Port C bits 2-5 output bits are forced to the push-pull output configuration for high speed data transmission, while pins set to input have the pull-up resistor disconnected, regardless of the state of PUPC.
The default condition of open drain output (SPI not enabled) allows software emulation of communication using the I^2C-bus protocol.

PORT D

The I/O pins of Port D normally are used for the input and output of video synchronization signals to the Sync Processor, but are set to I/O Input with pull-up upon reset. The I/O mode can be set individually for each port bit to Input with pull-up and output push-pull through the Port D DDR.

The configuration to support the Sync Processor required that the SYNOP bit of the PCR be reset to '0'; this enables Port D bits 0, 1 and 2 to the sync inputs and outputs.

Note that as these inputs are switched from normal I/O functionality, the video synchronization signals may also be monitored directly through the Port D Data Register for such tasks as checking for the presence of video signals or checking the polarity of Horizontal and Vertical synchronization signals (when the Sync Inputs are switched directly to the outputs using the multiplexors of the Sync Processor).

Table 6. I/O Pin Functions

R/$\overline{W}$*	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output the I/O pin.
1	0	The state of the I/O pin to read.
1	1	The I/O pin is in an output mode. The output data latch is read.

* R/$\overline{W}$ is an internal signal.

I/O PORTS (Continued)

2.2.3 Register Description

DATA REGISTERS

Port A: 0000h
Port B: 0001h
Port C: 0002h
Port D: 0003h

Read/Write

Reset Value: Undefined

7							0
MSB							LSB

DATA DIRECTION REGISTERS

Port A: 0004h
Port B: 0005h
Port C: 0006h
Port D: 0007h

Read/Write

Reset Value: 0000 0000 (00h) (as inputs)

7							0
MSB							LSB

PORT B CONFIGURATION REGISTER (0045h)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

b7-0 = **AD7-AD0:** *Port B Digital/Analog Input Configuration Bits.* When AD#i is set (i = 7-0), the pull-up on the respective pin #i of Port B is disconnected and the pin is configured as analog input; otherwise the pull-up is connected and pin configured as digital input (RESET condition) with no power consumption in the A/D channel.

Note. On the 42-pin package option, PB7-PB4 are not externally connected and are internally configured as standard digital inputs with pull-up.

PROGRAMMABLE INPUT/OUTPUT CONFIGURATION REGISTER (0046h)

Programmable Input/Output Configuration

Read/Write

Reset Value: 1111 1000 (F8h)

7							0
Res	CLMOP	OCOP	SYNOP	PUPC	POC2	POC1	POC0

b7 = **Res:** Reserved.

b6 = **CLMOP:** *Clamping Signal Output Select.* This bit selects either the PD3 I/O Pin Option or the output of the Clamping signal.

CLMOP = 0 : Clamping Signal

CLMOP = 1 : PD3 as Pull-up Input or Push-pull Output

b5 = **OCOP:** *Timer Output Compare Select.* This bit selects either the PC0 I/O Pin Option or the output of the Timer Output Compare.

OCOP = 0: Timer Output Compare

OCOP = 1: PC0 Input (with/no pull-up) or Output (push-pull)

b4 = **SYNOP:** *SYNC Processor Function Select.* This bit selects either the Sync Processor or PD0, PD1, PD2 I/O Pin Options.

SYNOP = 0: PD0 = CSYNCI, PD1 = HSYNCO and PD2 = VSYNCO

SYNOP = 1: PD0/PD1/PD2 Inputs (Pull-up) OR Outputs (Push-Pull).

Note. HSYNCO and VSYNCO can be directly read as Port bits by configuring PD1 and PD2 as inputs.

b3 = **PUPC:** *PORT C Input Configuration Bit.* This bit selects the input configuration for present bits of I/O Port C.

PUPC = 1 : Pull-up

PUPC = 0 : No Pull-up.

Whenever the SPI is active, the pull-up is disconnected from SPI input pins regardless of the state of PUPC and SPI outputs are set to push-pull.

b2-b0 = **POC2-POC0:** *PWM/BRM Output Configuration Bits.* These bits select the PWM/BRM output configuration.

PWM Group Channels		Value	
		0	1
A2 DA1, D3-6	P0C0	push-pull	open drain
B1 DA7-11	P0C1	push-pull	open drain
B2 DA12-DA17	P0C2	push-pull	open drain

In the case of uncomplete ports (Port C and Port D), non-implemented bits are read '0' whenever accessed.

2.3 16 BIT TIMER

2.3.1 Introduction

The 16-bit programmable timer consists of a 16-bit free running counter driven by a prescaler and control logic for two input captures and two output compare registers. It can be used for many purposes including pulse length measurement of input signals and generation of one output waveform.

The two input capture functions are dedicated to the timing functions of the Sync Processor and are internally connected to this source. They are thus not available for timing of other external signals.

When used with an 8MHz external oscillator frequency, the timer has a resolution of 0.5µs.

2.3.2 Functional Description

As the timer has a 16-bit architecture, each of its specific function blocks is represented by two registers. These registers contain the high order byte and low order byte of that function. However any access to the high order byte inhibits that specific timer capability until the low order byte is also accessed.

Note that correct software procedures should set the I bit of the Condition Code Register before accessing the high order byte to prevent an interrupt from occurring between the accesses to the high and low order bytes of any register.

Counter. The key element of the programmable timer is a 16-bit free running counter or counter register. It is preceded by a prescaler which divides the internal clock by two giving an operational frequency of 2MHz.

Software can read the counter at any time without affecting its value. It can be read from two locations, the Counter Register (0018h, 0019h) and Alternate Counter Register (001Ah, 001Bh). The only difference between these two read-only registers is the way the overflow flag TOF is handled during a read sequence.

A read sequence containing only a read of the least significant byte of the free running counter (from either the Counter Register or the Alternate Counter Register) will receive the LSB of the count value at the time of the read. A read of the most significant byte (from either the Counter Register or the Alternate Counter Register) simultaneously returns the MSB of the count value and causes the LSB to be transferred into a buffer.

The buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MSB several times. The read sequence is completed by reading the free running counter LSB, which actually returns the buffered value.

As shown in Figure 17 the free running counter is configured to FFFCh during reset, after $\overline{\text{RESET}}$ goes high. During a power-on reset (POR), the counter is also configured to FFFCh and begins running after the oscillator startup delay.

When the counter rolls over from FFFFh to 0000h, the Timer Overflow flag (TOF) of the Timer Status Register (TSR) is set. A timer interrupt is then generated if the TOIE enable bit of the Timer Control Register (TCR) is set, provided the I bit of the CCR is cleared. If one of these conditions is false, the interrupt remains pending to be issued as soon as they are both true. The interrupt request is cleared by reading TSR while TOF is set followed by an access (read or write) to the LSB of the Counter Register.

The TOF flag is not affected by accesses to the Alternate Counter Register. This feature allows simultaneous use of the overflow function and reads of the free running counter at random times (for example, to measure on elapsed time) without risking the clearing of the TOF flag erroneously. Accesses to the timer without the intention of servicing the TOF flag should therefore be performed to the Alternate Counter Register while only the TOF service routine accesses the Counter Register.

The free running counter can be reset under software control. This is performed by writing to the LSB of either the Counter Register or the Alternate Counter Register. The counter and the prescaler are then configured to their reset conditions. This reset also completes any 16-bit access sequence. All flags and enable bits are unchanged.

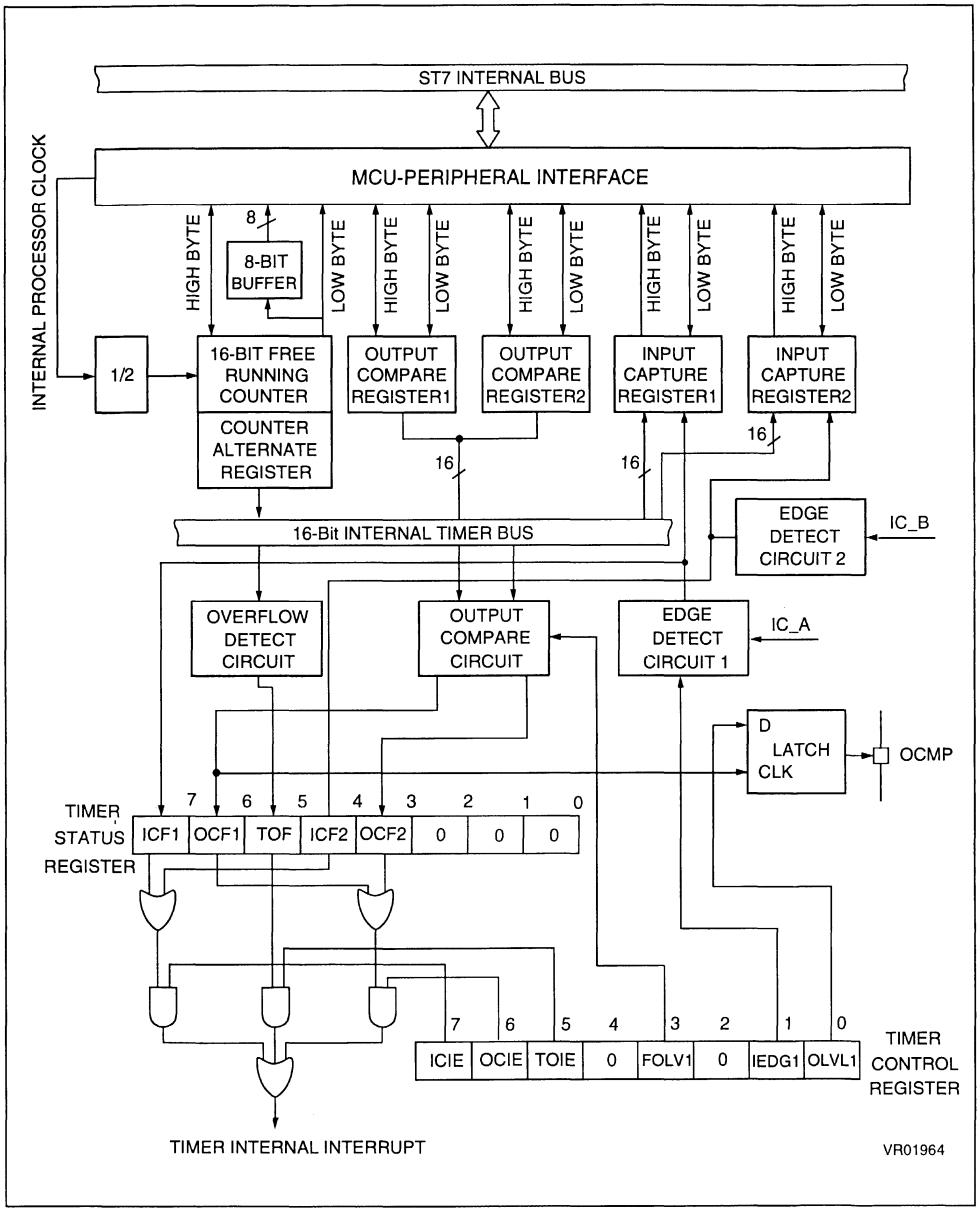
The value in the counter registers repeats every 131072 internal processor clock cycles (32 ms for f_{INT} = 4 MHz). As shown in Figure 17, the counter increment is triggered by a falling edge of the CPU clock.

The timer is not affected by the WAIT mode. In the HALT mode, the counter stops counting until the mode is exited. Counting then resumes from previous count (MCU woken by an interrupt) or from reset count (MCU woken by a reset).

Input Capture.

16 BIT TIMER (Continued)

Figure 15. Timer Block Diagram



16 BIT TIMER (Continued)

The ST7271 features two input capture registers and two input capture interrupt enable bits. The input capture inputs IC_A and IC_B are connected through the VSYNCI and HSYNCI (OR CSYNCI) input pins respectively. When the SYNC processor is not being used these pins may be used for the external input captures of the timer. The input on HSYNCI may optionally be passed through a /256 prescaler before being passed to the IC_B input capture.

Input Capture Register 1 (ICR1) is a 16-bit register made up of two 8-bit registers: the most significant byte register (ICHR1), located at 0014h, and the least significant byte register (ICLR1) located at 0015h.

ICR1 is a read-only register used to latch the value of the free running counter after a defined transition is sensed by the input capture edge detector at IC_A. This transition is software programmable through the IEDG1 bit of the Timer Control Register (TCR). When IEDG1 is set, a rising edge triggers the capture; when IEDG1 is low, the capture is triggered by a falling edge. Care must be taken with the external circuitry to avoid unwanted interrupts when changing the interrupt edge.

When an input capture occurs, the flag ICF1 in Timer Status Register (TSR) is set. An interrupt is requested if the interrupt enable bit ICIE of TCR is set, provided the I bit of the CCR is cleared. Otherwise, the interrupt remains pending until both conditions become true. It is cleared by reading the Timer Status Register TSR followed by an access (read or write) to the LSB of ICR1.

The result stored in ICR1 is one more than the value of the free running counter on the rising edge of the internal processor clock preceding the active transition at pin IC_A (see Figure 17). This delay is required for internal synchronization. Therefore, the timing resolution of the input capture system is one count of the free running counter, i.e. 2 internal clock cycles.

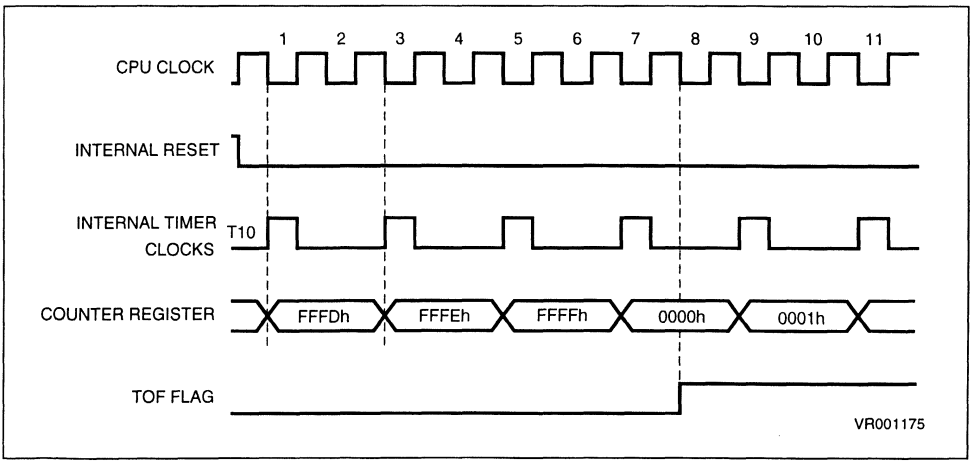
The free running counter is transferred to ICR1 on each proper signal transition regardless of whether the Input Capture Flag ICF1 is set or cleared. The ICR1 always contains the free running counter value which corresponds to the most recent input capture.

After a read of the MSB of ICR1 (ICHR1), counter transfer of input capture is inhibited until the LSB of ICR1 (ICLR1) is also read. This characteristic forces the minimum pulse period attainable to be determined by the time to service the interrupt and to execute the interrupt routine.

A read of ICLR1 does not inhibit the counter transfer. Again, minimum pulse periods are the ones which allow software to read the least significant byte and perform needed operations. There is no conflict between the read of ICR1 and the running counter transfer since they occur on opposite edges of the internal processor clock (see Figure 17).

The ICR1 is undetermined at power-on and is not affected by an external reset. Hardware circuitry has to provide protection from generating a wrong input capture when changing the edge sensitivity option of IC_A input through the IEDG1 bit.

Figure 16. Timer Timing Diagram



16 BIT TIMER (Continued)

During the HALT mode, if at least one valid input capture edge occurs at the IC_A input, the input capture detect circuitry is armed. This action does not set any timer flags nor "wake-up" the MCU. If the MCU is awaken by an interrupt, there is an active input capture flag and data from the first valid edge that occurred during the HALT mode. If the HALT mode is exited by a reset, the input capture detect circuitry is reset and thus, any active edge that happened during the HALT mode is lost.

Input Capture Register 2 (ICR2) is a 16-bit register made up of two 8-bit registers: the most significant byte register (ICHR2), located at 001Ch, and the least significant byte register (ICLR2) located at 001Dh.

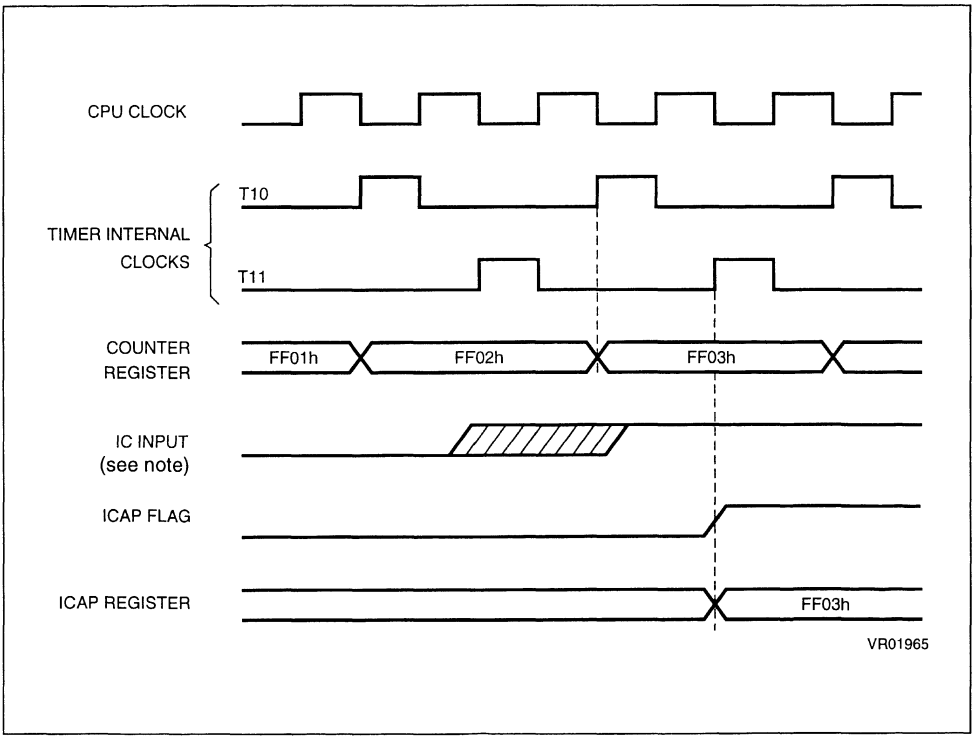
The previous description shown for Input Capture Register 1 is also applicable for the Input Capture Register 2, with the exception that Input Capture Register 2 is triggered only on a negative edge on input IC_B and with the substitution of the appropriate index in the bit and register names.

Output Compare.

There are two output compare registers: Output Compare Register 1 and 2 (OCR1 and OCR2). They can be used for several purposes such as controlling an output waveform or indicating when a period of time has elapsed. OCMP1 pin is associated with output compare 1; no pin is associated with Output Compare 2 which can be used for the generation of timer interrupts.

The Output Compare Registers are unique because all bits are readable and writable and are not affected by the timer hardware and reset. If a compare function is not used, the two bytes of the corresponding Output Compare Registers can be used as storage locations. Note that the same output compare interrupt enable bit is used for both output compares.

Figure 17. Input Capture Timing Diagram



Note. The diagram represents the case of a rising edge sensitivity (IEDG1=1). The capture is performed at the next rising edge of T11, if the action edge of ICAP happened before the previous T10 falling edge.

16 BIT TIMER (Continued)

Output Compare Register 1. The Output Compare Register 1 (OCR1) is a 16-bit register, which is made up of two 8-bit registers: The most significant byte register (OCHR1) at address 0016h and the least significant byte register (OCLR1) at address 0017h.

The content of OCR1 is compared with the content of the free running counter once during every timer clock cycles, i.e. every 2 internal processor clock periods. If a match is found, the Output Compare Flag OCF1 of the TSR is set and the Output Level bit (OLVL1) of the TCR is clocked to the OCMP1 pin (see output compare timing diagram on Figure 19).

OLVL1 is copied to the corresponding output level latch and hence, to the OCMP1 pin regardless of whether the Output Compare Flag (OCF1) is set or not. The value in the OCR1 and the OLVL1 bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

An interrupt follows a successful output compare if the corresponding interrupt enable bit OCIE of the TCR is set, provided the I-bit of the CCR is cleared. Otherwise, the interrupt remains pending until both conditions are true. It is cleared by a read of TSR followed by an access to the LSB of the OCR1.

After a processor write cycle to the OCHR1 register, the output compare function is inhibited until the OCLR1 is also written. Thus, the user must write both bytes if the MSB is written first. A write made to only the LSB will not inhibit the compare function. The minimum time between two successive edges on the OCMP1 pin is a function of the software program.

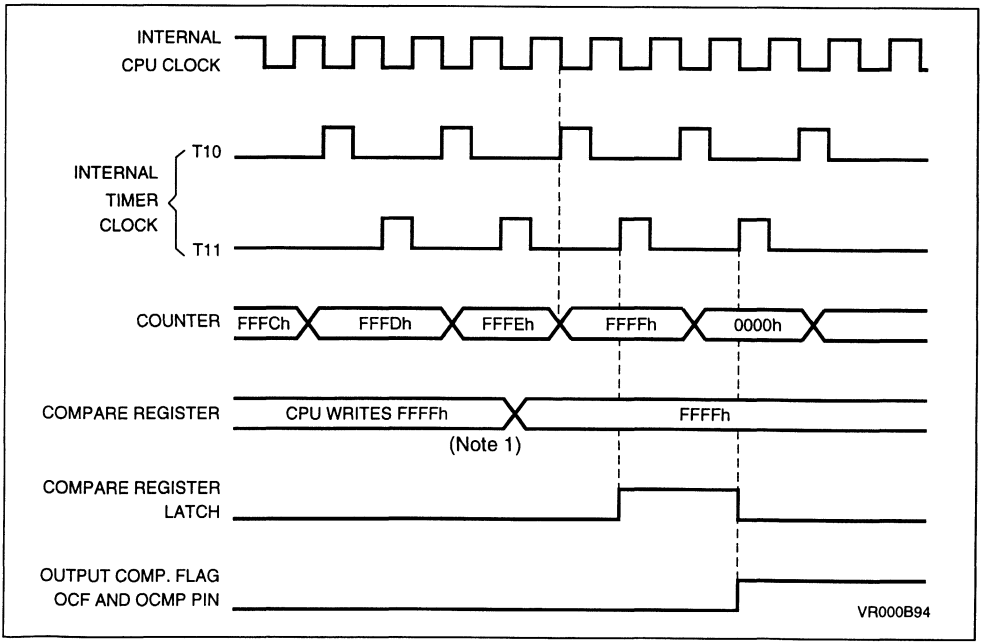
The OCMP1 output latch is forced low during reset and stays low until valid compares change it to a high level. Because the OCF1 flag and the OCR1 are undeterminate at power-on and are not affected by an external reset, care must be exercised when initiating the output compare function with software. The following procedure is recommended to prevent the OCF1 flag from being set between the time it is read and the write to OCR1:

Write to OCHR1 (further compares are inhibited).

Read the TSR (first step of the clearance of OCF1 [it may be already set]).

Write to OCLR1 (enables the output compare function and clears OCF1).

Figure 18. Output Compare Timing Diagram



Note 1. The CPU write to the compare registers may take place at any time but a compare only occurs at timer state T11. Thus a 2-cycles difference may exist between the write to the compare register and the actual compare.

16 BIT TIMER (Continued)

Output Compare Register 2. The Output Compare Register 2 (OCR2) is a 16-bit register, which is made up of two 8-bit registers: the most significant byte register (OCHR2) at address 0001Eh and the least significant byte register (OCLR2) at address 0001Fh.

This register works as the Output Compare Register 1. For a complete description, please refer to the above and substitute the appropriate index in the bit and register names.

Software Force Compare. The force compare capability main purpose is to facilitate fixed frequency generation.

When the Force Output Level 1 bit (FOLV1) of TCR is written to 1, OLVL1 is copied to pin OCMP1. To provide this capability, internal logic allows a single instruction to change OLVL1 and causes a forced compare with the new value of OLVL1. OCF1 is not affected and thus, no interrupt request is generated.

2.3.3 Register Description

TIMER CONTROL REGISTER (0012h)

Read/Write
Reset Value: 0000 00x0 (00h or 02h)

7							0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1

b7 = **ICIE** *Input Capture Interrupt Enable*
If ICIE is set, a timer interrupt is enabled whenever the ICF1 status flags of TSR are set. If the ICIE bit is cleared, the interrupt is inhibited.

b6 = **OCIE:** *Output Compare Interrupt Enable*
If OCIE is set, a timer interrupt is enabled whenever the OCF1 or OCF2 status flags of TSR are set. If the OCIE bit is cleared, the interrupt is inhibited.

b5 = **TOIE:** *Timer Overflow Interrupt Enable*
If TOIE is set, a timer interrupt is enable whenever the TOF status flag of TSR is set. If the TOIE bit is cleared, the interrupt is inhibited.

b4 = **FOLV2:** *Force Output Compare 2*
This bit has no affect. FOLV2 is cleared by a system reset.

b3 = **FOLV1:** *Force Output Compare 1*
When written to 1, FOLV1 forces OLVL1 to be copied to the OCMP pin.

b2 = **OVL2**: *Output Level 2.* This bit has no affect.

b1 = **IEDG1:** *Input Edge 1*
The value of the IEDG1 determines which level transition on IC_A input will trigger a free running counter transfer to the ICR1. When IEDG1 is high, a rising edge triggers the capture since when low, a falling edge does.

b0 = **OLVL1** *Output Level 1*
The OLVL1 bit is copied to the OCMP1 pin whenever a successful comparison occurs at OCR1.

TIMER STATUS REGISTER (0013h)

Read Only
Reset Value: Undefined

7							0
ICF1	OCF1	TOF	ICF2	OCF2	0	0	0

b7 = **ICF1** *Input Capture Flag 1*
ICF1 is set when a proper edge has been sensed by the input capture edge detector at IC_A. The edge is selected by the IEDG1-bit in TCR. ICF1 is cleared by a processor access to the TSR while ICF1 is set followed by an access (read or write) to the low byte of ICR1 (ICLR1).

b6 = **OCF1** *Output Compare Flag 1*
OCF1 is set when the content of the free running counter matches the content of OCR1. It is cleared by a processor access of TSR while OCF1 is set followed by an access (read or write) to the low byte of OCR1.

b5 = **TOF** *Timer Overflow*
TOF is set by a transition of the free running counter from FFFFh to 0000h. It is cleared by a processor access to TSR while TOF is set followed by an access (read or write) to the low byte of the counter low register. TOF is not affected by an access to the Alternate Counter Register.

b4 = **ICF2** *Input Capture Flag 2*
ICF2 is set when a negative edge has been sensed by the input capture edge detector at IC_B. ICF2 is cleared by a processor access to the TSR while ICF2 is set followed by an access (read or write) to the low byte of ICR2 (ICLR2).

b3 = **OCF2** *Output Compare Flag 2*
OCF2 is set when the content of the free running counter matches the content of OCR2. It is cleared by a processor access of TSR while OCF2 is set followed by an access (read or write) to the low byte of OCR2.

b2-0 = **Unused**, read as '0'.

16 BIT TIMER (Continued)

INPUT CAPTURE REGISTER 1, High Byte (0014h)

Read Only
Reset Value: Undefined

7							0
IC1.15	IC1.14	IC1.13	IC1.12	IC1.11	IC1.10	IC1.9	IC1.8

INPUT CAPTURE REGISTER 1, Low byte (0015h)

Read Only
Reset Value: Undefined

7							0
IC1.7	IC1.6	IC1.5	IC1.4	IC1.3	IC1.2	IC1.1	IC1.0

OUTPUT COMPARE REGISTER 1, High byte (0016h)

Read/Write
Reset Value: Undefined

7							0
OC1.15	OC1.14	OC1.13	OC1.12	OC1.11	OC1.10	OC1.9	OC1.8

OUTPUT COMPARE REGISTER 1, Low byte (0017h)

Read/Write
Reset Value: Undefined

7							0
OC1.7	OC1.6	OC1.5	OC1.4	OC1.3	OC1.2	OC1.1	OC1.0

COUNTER REGISTER, High byte (0018h)

Read Only
Reset Value: 1111 1111 (FFh)

7							0
C1.15	C.14	C.13	C.12	C.11	C.10	C.9	C.8

COUNTER REGISTER, Low byte (0019h)

Read/Write
Reset Value: 1111 1100 (FCh)

7							0
C.7	C.6	C.5	C.4	C.3	C.2	C.1	C.0

Writing to this Register will cause the counter to be reset to its reset value of FFFCh. Flags and enable bits remain unaltered by this operation.

ALTERNATE COUNTER REGISTER
High byte (001Ah)

Read Only
Reset Value: 1111 1111 (FFh)

7							0
AC.15	AC.14	AC.13	AC.12	AC.11	AC.10	AC.9	AC.8

ALTERNATE COUNTER REGISTER
Low byte (001Bh)

Read/Write
Reset Value: 1111 1100 (FCh)

7							0
AC.7	AC.6	AC.5	AC.4	AC.3	AC.2	AC.1	AC.0

Writing to this Register will cause the counter to be reset to its reset value of FFFCh. Flags and enable bits remain unaltered by this operation.

INPUT CAPTURE REGISTER 2, High byte (001Ch)

Read Only
Reset Value: Undefined

7							0
IC2.15	IC2.14	IC2.13	IC2.12	IC2.11	IC2.10	IC2.9	IC2.8

INPUT CAPTURE REGISTER 2, Low byte (001Dh)

Read Only
Reset Value: Undefined

7							0
IC2.7	IC2.6	IC2.5	IC2.4	IC2.3	IC2.2	IC2.1	IC2.0

OUTPUT COMPARE REGISTER 2, High byte (001Eh)

Read/Write
Reset Value: Undefined

7							0
OC2.15	OC2.14	OC2.13	OC2.12	OC2.11	OC2.10	OC2.9	OC2.8

OUTPUT COMPARE REGISTER 2, Low byte (001Fh)

Read/Write
Reset Value: Undefined

7							0
OC2.7	OC2.6	OC2.5	OC2.4	OC2.3	OC2.2	OC2.1	OC2.0

2.4 SYNC PROCESSOR

2.4.1 Introduction

The Sync Processor handles all the management tasks of the video synchronisation signals, and is used with the Timer and software to provide information and status on the video standard and timings.

Separated Horizontal and Vertical Synchronization pulses, provided on the HSYNCI and VSYNCI pins, are accepted, with polarity detection by software. In this case HSYNCO = HSYNCI (with programmable polarity inversion), without any blanking.
Alternatively a composite sync signal (sync pulses only) may be provided on CSYNCI (or the HSYNCI pin), with automatic synchronisation pulse extraction (with polarity detection by software).

Extraction of VSYNCO may be made from a composite signal (OR, XOR or serration combinations of Horizontal and Vertical components).
Note. If the input is a composite signal both VSYNCO and HSYNCO will be extracted (the latter blanked during VSYNCO pulse as far as potential serration pulses are concerned).

Processed sync pulses may be output to external parts of the circuit through the HSYNCO and VSYNCO pins with programmable polarity. An independent programmable-duration back-porch (clamping) output signal (CLMPO) may also be combined externally to extend the Horizontal sync output on HSYNCO. In the case of extraction of HSYNCO and VSYNCO from CSYNCI this signal is suppressed during vertical blanking.

2.4.2 Functional Description

The function of the Sync Processor can be summarized as the 4 following tasks:

- Check the presence of input signals (VSYNCI, HSYNCI and CSYNCI)
- Polarity Detection (VSYNCI, HSYNCI and CSYNCI)
- HSYNCO, VSYNCO Extraction
- Video Standard Discrimination

These tasks are performed by the Sync Processor in close conjunction with the Timer, and user software.

The block diagram of the Sync Processor is shown in Figure 20. This also shows the internal connections to the Timer Input Capture A (IC_A) and Input Capture B (IC_B).

Checking the presence of input signals

The Sync Processor offers two techniques for checking the prescence of signals, a software intensive direct check and an indirect check which is interrupt driven and uses the hardware of the Sync Processor. This second technique thus offers more time for other tasks.

Direct Check

The direct check is made by monitoring the input status directly on the I/O pins. To do this, the internal multiplexors of the Sync Processor are set to allow the direct pass-through of the incoming synchronization signals through to the corresponding output:

HSYNCI => HSYNCO (PD1)

VSYNCI => VSYNCO (PD2)

The corresponding sync output pins are set to the normal I/O mode (by setting the SYNOP bit of PIOCR) and the state of the inputs are read over a period of time in order to detect any transitions on the input. If found it can be assumed that signals are present.

CSYNCI can be monitored by using the PD0 input function directly.

Indirect Check

To use the indirect check, the multiplexers are set to connect the VSYNCI input to Timer Input Capture A and and HSYNCI or CSYNCI to the Input Capture B.

step I - Checking VSYNCI. Any interrupt request coming from IC_A is monitored. (On detecting VSYNCI, the software may either detect the VSYNCI polarity or check for the presence of HSYNCI).

step II - Checking HSYNCI. The input HSYNCI is connected directly to IC_B. An interrupt request is waited for (on detecting HSYNCI, the software may either detect the HSYNCI polarity or check the CSYNCI presence).

step III - Checking CSYNCI. The CSYNCI input is connected directly to IC_B. An interrupt request is waited for.

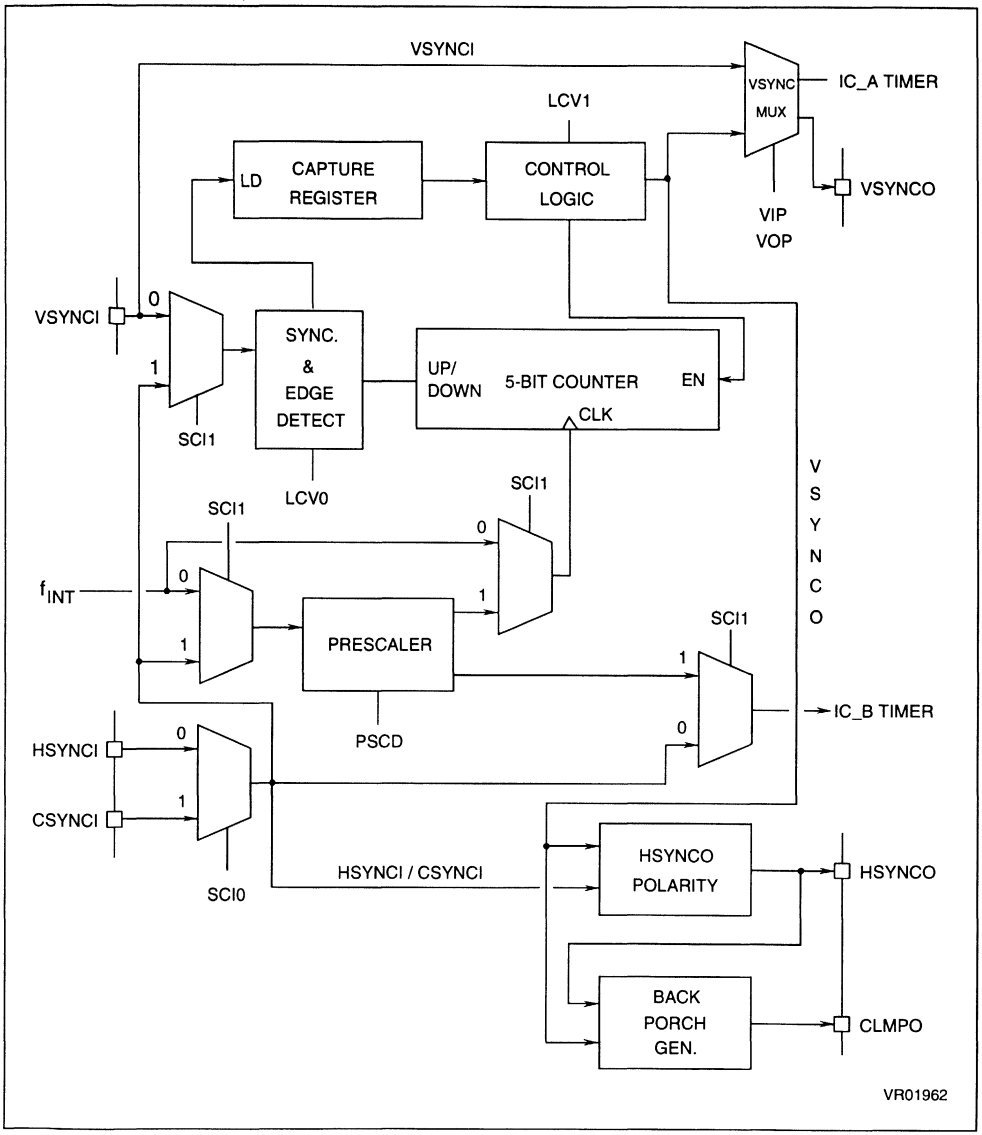
Note. Input Capture A edge detection polarity may be selected to be positive or negative.

Input Capture B edge detection polarity is fixed to negative edges.

Steps I-III may be carried out in parallel.

SYNC PROCESSOR (Continued)

Figure 19. Sync Processor Block Diagram



SYNC PROCESSOR (Continued)

Polarity Detection.

The Sync Processor again offers two techniques for checking the polarity of signals, a software intensive direct check and an indirect check which uses the hardware of the Sync Processor. This second technique thus offers more time for other tasks.

Indirect Polarity Detection

To check signal polarity with the indirect method, the internal 5-bit up/down counter is used. At the beginning of the detection phase, '11111' is written into the SYNC control register (CVM bits). These bits are updated by the 5-bit counter value at every detected edge (for example: positive) on the signal considered. The counter increments when the signal is high; otherwise it decrements.

Software can thus check the SYNC Processor capture register after an interrupt (with the signal connected to IC_A or IC_B) or by polling. In case of a positive polarity, the capture value will be '00000' as the counter stays at this value after underflowing. Otherwise, it will be different to '00000' (assured by the resolution of the counter) and thus be negative polarity.

This one-shot detection approach covers separate HSYNCI and VSYNCI signals only. In case of a composite incoming signal, the software should consider reading the Sync Processor register every 7 or 8 interrupt requests, or a number of lines to ensure that the readings are outside the vertical blanking period and to minimize the possibility of misleading polarity values.

Direct Polarity detection

The alternative, software intensive, technique is to read the status of the VSYNCI and HSYNCI signals through the I/O pins as summarized in the previous task description.

An average value can be determined by counting (in software) the number of times the level is '1' and the number of times the level is '0' over a period lasting several frame or line periods or by repeating the counting several times over an equal period. The total values can then be compared and the greater of the counts between '1' and '0' indicates the signal polarity.

Figure 20. Horizontal Sync. Input Timing

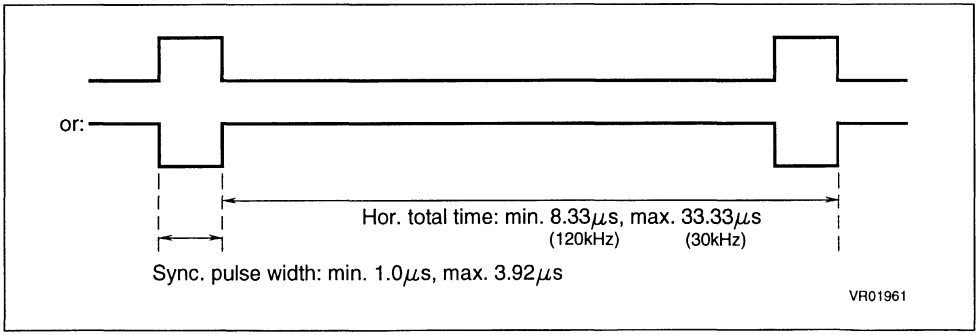
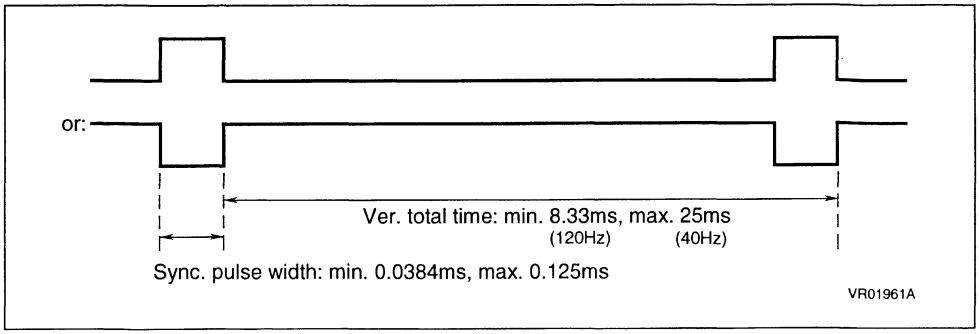


Figure 21. Vertical Sync. Input Timing



SYNC PROCESSOR (Continued)

VSYNCO Extraction

VSYNCO is extracted from CSYNCI with the aid of the 5-bit up/down counter. Initially, the width of a Horizontal sync component pulse is determined automatically by the hardware, which defines a threshold for the counter (which may be replaced with an optional user defined value including a tolerance factor), the circuit then monitors for any incoming period greater than this captured value. This is then processed as the VSYNCO signal.

The user software should first select the acquisition mode to measure the internal Horizontal sync component pulse width. The time-equivalent value is read after this value is captured in the internal register. If a user-defined tolerance is to be added, then an updated value can be re-written into the register.

The capture occurrence can be indicated by the timer Input Capture interrupt or noted by reading a new value on the Sync Processor Control Register. After this step, the software should set the extraction mode to continue the VSYNCO generation by hardware as shown in the following paragraph for a negative polarity signal.

In extraction mode, the 5-bit comparator checks the counter value with respect to the threshold. When the counter reaches the threshold on its way down, VSYNCO is asserted. During the vertical blanking, counter value is decreased until it reaches a programmable minimum, i.e. it does not underflow. When the vertical period is finished, the counter starts counting up and when the maximum is reached, VSYNCO is negated. The extracted signal may be validated by software since it is input to Timer IC_A.

The threshold is greater than the count for a HSYNCI pulse. Serration pulses during vertical blanking are thus filtered out. Similarly, positive CSYNCI signals are covered by properly selecting the edge sensitivity on the Hsync-width-measurement mode.

Standard Discrimination

Discrimination of video standards is supported by software and the Timer used in conjunction with the Sync Processor. For this purpose, either HSYNCI or CSYNCI is prescaled by a fixed factor of 256.

Note. the prescaler may be bypassed for frequency sychronization signals lower than specified.

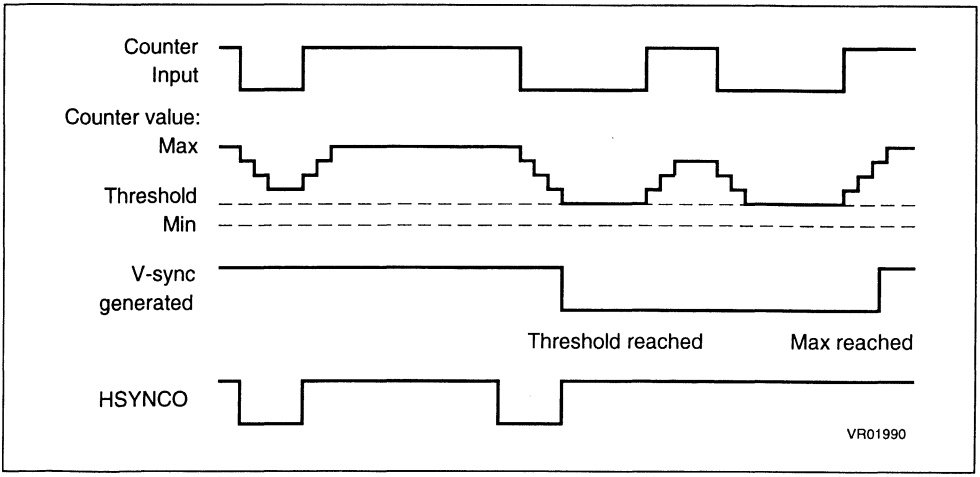
The function is carried out by using the Timer Input-Capture Channels:

a) VSYNCI is directly connected to IC_A (after synchronization).

b) HSYNCI or CSYNCI is prescaled (or not) by a factor of 256 in the Sync Processor and then sent to IC_B.

Signal timing can be directly calculated by the time values between the appropriate interrupts given by the Timer, and then used for comparison against existing pre-defined standards. It is recommended that several captured values are averaged to give a more accurate timing for comparison if serration pulses are present on CSYNCI.

Figure 22. VSYNCO generation for a composite signal (+ serration pulses) with a negative polarity



SYNC PROCESSOR (Continued)

Inputs

The inputs to the Sync Processor are the Video Synchronization strobe pulses:

VSYNCI (Vertical Sync input, TTL Level, Schmitt triggered).

HSYNCI (Horizontal Sync input, TTL Level, Schmitt triggered).

CSYNCI (Composite Sync input, TTL Level, Schmitt triggered).

NOTE: The Composite Sync signal may also be received on the HSYNCI input if this is supplied by the external circuit and the I/O function of the corresponding I/O pin for CYSYNCI is required.

Input Signal Waveforms.

The input signals must contain only synchronization pulses.

Timing characteristics of HSYNCI and VSYNCI

HSYNCI:

In case of serration pulses on CSYNCI/HSYNCI, these pulses should be externally generated with a minimum half-a-line delay from the VSYNCI edge.

The HSYNCI or CSYNCI signal, optionally prescaled by 256, is connected to the IC_B input (Timer Input Capture B) of the Timer.

The Timer resolution is 500ns for an external oscillator frequency of 8MHz.

Outputs

HSYNCO: HSYNC Output, (CMOS Level). With programmable polarity, this signal is blanked during the vertical period (if the input is a composite signal). Its internal propagation delay has been optimised to its lowest possible delay.

If separated HSYNCI and VSYNCI are provided, no blanking is generated on HSYNCO.

VSYNCO: VSYNC Output, (CMOS Level) with programmable polarity.

VSYNCO is connected to IC_A, input to Timer Input Capture A. The input to Timer is delayed by 125ns-250ns by synchronization with the internal clock.

CLMPO: (CMOS level) back porch clamp signal.

If VSYNCO is extracted from a composite signal, the minimum delay is 500ns + HSYNCO pulse width. The maximum delay is software defined (the threshold value on extraction mode) and corresponds to 8750 ns.

Notes

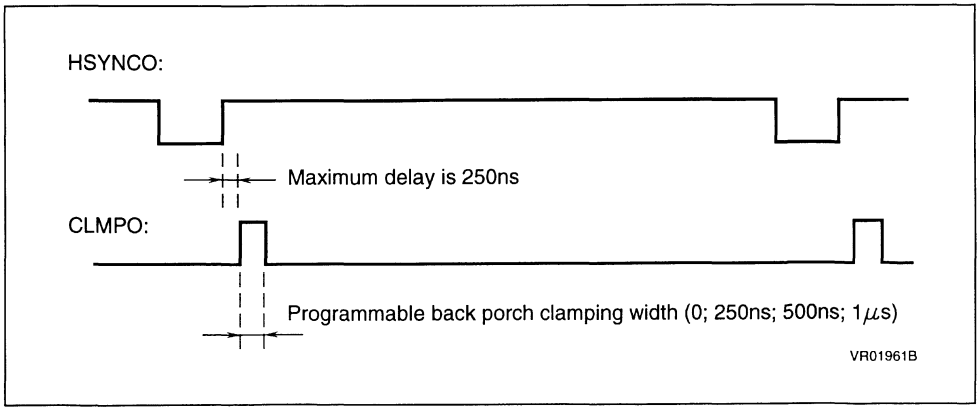
- Standards with less than 256 lines per frame are NOT supported.
- If separated HSYNCI and VSYNCI are provided, no blanking is generated on HSYNCO.
- The following are not supported:

  A back-porch clamp signal generator on VSYNCO.

  A front-porch clamp signal generator.

  Pre/post-equalizing pulses.
- No direct interrupt request is used by the Sync Processor, although the optional interrupt in the timer can be used by the software since VSYNCO and HSYNCI/CSYNCI signals are connected to Input Capture (IC_A and IC_B, respectively).
- The Timer Interrupt Request should be masked during a write access to the SYNC Control Registers.

Figure 23. Back Porch (CLMPO) Delay



SYNC PROCESSOR (Continued)

2.4.3 Register Description

Two 8-bit read/write registers are used to control the Sync Processor:

- Counter Control Register (CCR)
- Mux Control Register (MCR)

COUNTER CONTROL REGISTER (003Dh)

Reset Value 0000 0000 (00h)

Read/Write

7							0
PSCD	LCV1	LCV0	CV4	CV3	CV2	CV1	CV0

b7 = **PSCD:** *Prescaler Disable.*

- If set to '0' the 8-bit Prescaler is enabled.
- If set to '1' the 8-bit Prescaler is disabled and preset to 1111 1110b.

b6-5 = **LCV1,LCV0:** *VSYNCO Extraction Control Bits*

LCV1	LCV0	VSYNC0 Control Bits
0	0	Acquisition mode CSYNCI/HSYNCI Negative polarity Counter capture on input signal falling edge
0	1	Acquisition mode CSYNCI/HSYNCI Positive polarity Counter capture on input signal rising edge
1	0	Extraction mode CSYNCI/HSYNCI Negative polarity CV4-0 = counter minimum threshold
1	1	Extraction mode CSYNCI/HSYNCI Positive polarity CV4-0 = counter maximum threshold

b4-0 = **CV4-0:** *Counter Captured Value.* These bits correspond to the counter captured value in different modes. Upon VSYNCO extraction, it corresponds to a HSYNCI pulse-width measurement, which may be changed by software before extraction.

MUX CONTROL REGISTER (003Ch)

Reset Value: 0000 0000 (00h)

Read/Write

7							0
BP1	BP0	SCI1	SCI0	HS1	HS0	VOP	VIP

b7-6 = **BP1, 0:** *Back Porch (CLMPO) width control*

BP1	BP0	Back Porch Width
0	0	No Back Porch
0	1	250ns
1	0	500ns
1	1	1000ns

b5 = **SCI1:** *Horizontal/Vertical Signal Path Selection Bit.* This bit selects the path for the incoming signals towards IC_A, IC_B, VSYNCO and HSYNCO:

SCI1 = "0" if the task/objective is:

- VSYNCI Polarity Detection using counter
- IC_B connected to HSYNCI/CSYNCI without prescaling

SCI1 = "1" if the task/objective is:

- HSYNCI/CSYNCI Polarity Detection using counter
- HSYNCI/CSYNCI prescaled and connected to IC_B
- VSYNCO extraction

b4 = **SCI0:** *HSYNCI/CSYNCI Selection Bit.* This bit selects either HSYNCI/CSYNCI as input.

- If SCI0 = "0" -> HSYNCI pin is selected.
- If SCI0 = "1" -> CSYNCI pin is selected.

b3-2 = **HS1, HS0:** *Horizontal Signal Selection Bits.* These bits allow inversion of the HSYNCI/CSYNCI polarity, output as HSYNCO, as well as the generation of CLMPO as follows:

HS1	HS0	HSYNC Selection Mode
0	0	CLMPO after HSYNCO rising edge HSYNC0 <- (HSYNCI,CSYNCI)
0	1	CLMPO after $\overline{\text{HSYNCOI}}$ rising edge HSYNC0 <- ($\overline{\text{HSYNCI}}$, $\overline{\text{CSYNCI}}$)
1	0	CLMPO after $\overline{\text{HSYNCOI}}$ falling edge HSYNC0 <- ($\overline{\text{HSYNCI}}$, $\overline{\text{CSYNCI}}$)
1	1	CLMPO after HSYNCOI falling edge HSYNC0 <- (HSYNCI,CSYNCI)

b1-0 = **VOP,VIP:** *Vertical Signal Polarity Selection Bits.* These bits are only set/cleared by software and are related to the polarity of the incoming and the outgoing vertical signal.

VIP should be written to '0' by software after detection of a negative polarity on VSYNCI. Otherwise, written to '1'.

VOP should be written to '0' by software to select a negative polarity for VSYNCO. Otherwise, written to '1'.

VOP	VIP	VSYNC Selection Mode
0	0	ICAP_A <- VSYNCI VSYNCO <- VSYNCI
0	1	ICAP_A <- $\overline{\text{VSYNCI}}$ VSYNCO <- VSYNCI
1	0	ICAP_A <- VSYNCI VSYNCO <- $\overline{\text{VSYNCI}}$
1	1	ICAP_A <- $\overline{\text{VSYNCI}}$ VSYNCO <- $\overline{\text{VSYNCI}}$

2.5 DIGITAL TO ANALOG CONVERTER

PULSE-WIDTH MODULATOR (PWM) + BINARY-RATE MULTIPLIER (BRM)

2.5.1 Introduction

The ST7271 provides two types of Digital to Analog Converters with differing step resolutions based on the Pulse-Width Modulator (PWM) and Binary Rate Multiplier (BRM) Generator technique. These may act as digital potentiometers when used with external filtering to control such elements as brightness, saturation/contrast and other analog variables.

- A 10-Bit PWM/BRM with a repetition rate of 64KHz, 250ns resolution and a step of 5mV (0/5V, excepting DA2). In the 56-pin package, 16 channels are provided with this configuration (channels PWM2-PWM17 with outputs DA2-DA17 respectively). DA2 has a fixed open-drain output, with an external V_{DD} capability up to 12V, while DA3-DA17 are programmable to open-drain or push-pull output configuration, with a 0-V_{DD} (+5V) range.

- A 12-bit PWM/BRM generator (2 Channels: PWM0 and PWM1) with a repetition rate of 64KHz, 250ns resolution and a step of 1.25mV (0/5V excepting DA0). The channels PWM0 and PWM1 correspond with outputs DA0 and DA1 respectively. DA0 has a fixed open-drain output, with an external V_{DD} capability up to 12V, while DA1 is programmable to open-drain or push-pull output configuration, with a 0-V_{DD} (+5V) range.

2.5.2 Functional Description

10-BIT PWM/BRM

The 10-Bits of the 10-bit PWM/BRM are distributed as 6 PWM bits and 4 BRM bits. The generator consists of a 10-bit counter (common for all channels), a comparator and the PWM/BRM generation logic.

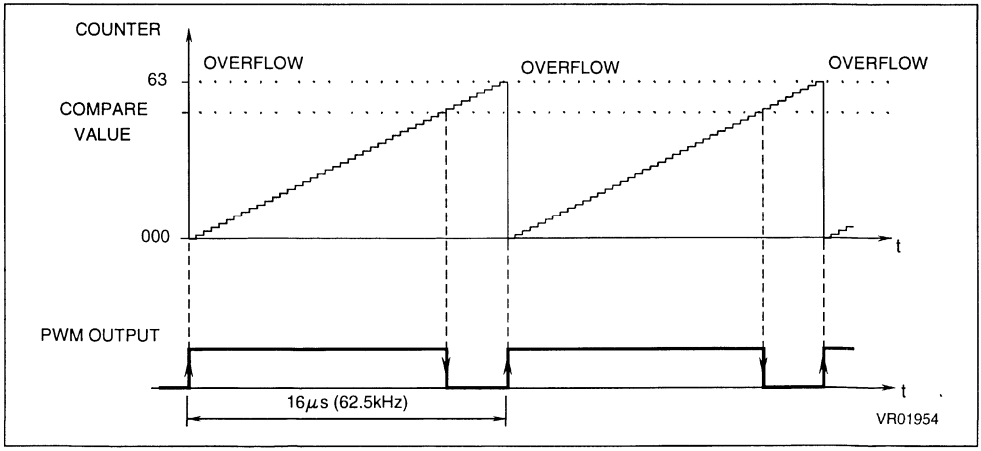
PWM Generation

The counter increments continuously, clocked at the Main oscillator frequency divided by 2 (with a period $t_{CLK} = 1/f_{CLK} = 250ns$). Whenever the 6-Least Significant bits of the counter (defined as the PWM counter) overflow, the output level for all active channels is set.
The state of the PWM counter is continuously compared to the PWM binary weight for each channel, software-defined in the relevant PWM register, and when a match occurs the output level for that channel is reset.

This Pulse Width modulated signal is to be filtered with an external RC network, placed as close as possible to the associated pin. This provides an analog voltage proportional to the average charge passed to the external capacitor. Thus for a higher mark/space ratio (High time much greater than Low time) the output voltage is higher. The external components of the RC network should be selected to give the optimum filtering level needed to control the system variable.

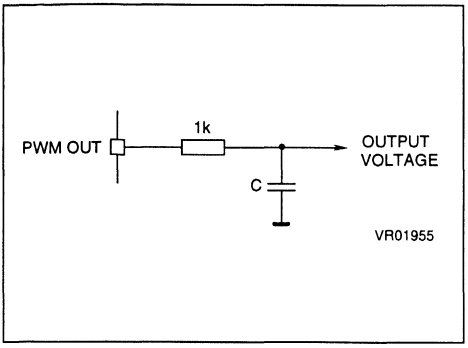
Each output may individually have its polarity inverted by software action.

Figure 24. PWM Generation



DIGITAL TO ANALOG CONVERTER (Continued)

Figure 25. Typical PWM Output Filter



6-Bit PWM Ripple After Filtering

C (µF)	V_{PP_RIPPLE} (mV)	τ (ms)
0.256	78	0.256
2.56	7.8	2.56
25.6	0.78	25.6

Assuming RC filter (R=1kΩ) and V_{DD} = 5V

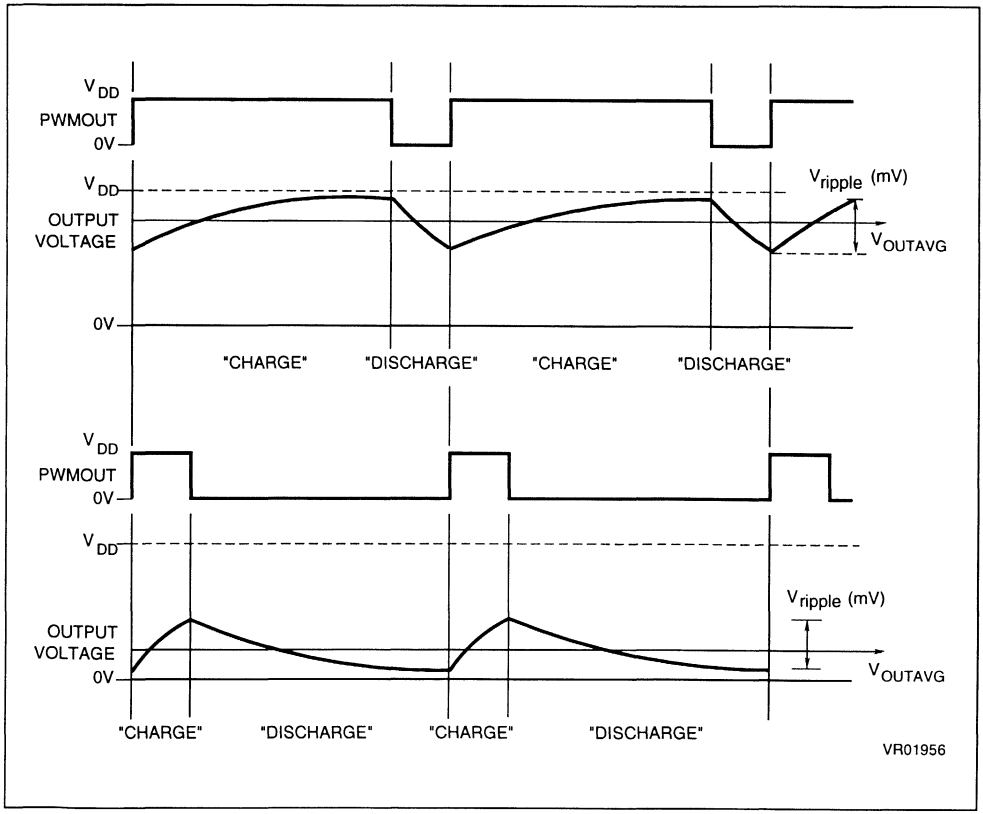
PWM Duty Cycle 50%

Step = 5V/64 = 78mV which requires a minimum

τ (filter time constant) of $\frac{250ns \times 64 \times 64}{4} = 256\mu s$

to ensure integral linearity of ±0.5LSB

Figure 26. PWM Simplified Voltage Output After Filtering (2 examples)



DIGITAL TO ANALOG CONVERTER (Continued)

BRM Generation

The BRM bits allow the addition of a 250ns-pulse to widen a standard PWM pulse at specific PWM cycles. This has the effect of "fine-tuning" the PWM Duty cycle (without modifying the base duty cycle), thus, with the external filtering, providing additional fine voltage steps.

The incremental pulses (with duration of t_{CLK} = $1/f_{CLK}$ = 250ns) are added to the beginning of the original PWM pulse. The PWM intervals which are added to are specified in the 4-bit BRM register and are encoded as shown in the following table. The BRM values shown may be combined together to provide a summation of the incremental pulse intervals specified.

The pulse increment corresponds to the PWM resolution. For example, if data 18h is written to the PWM register and data 06h (00000110b) to the BRM register, for a 4 MHz internal clock (250ns resolution), a 6.0 µs-long pulse will be output at every 64µs interval except at those numbered as #2,4,6,10,12,14 where the pulse is broadened to 6.25µs.

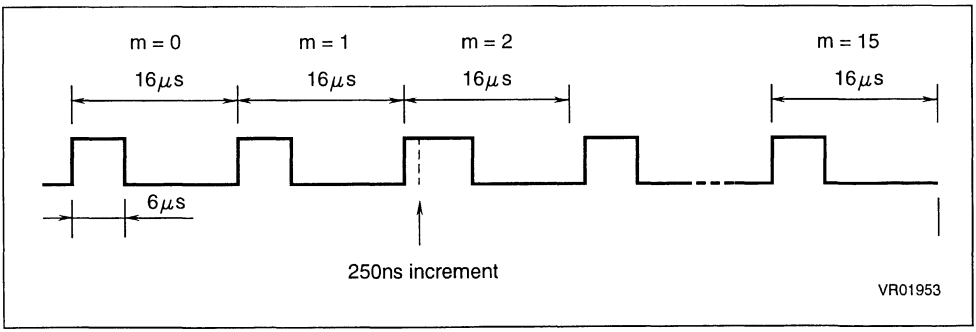
Note. If data 00h is written to both PWM and BRM registers, the generator output will remain at "0". Conversely, if both registers hold data 3Fh and 0Fh, respectively , the output will remain at "1" for all intervals #1 ≤ i ≤ #15, but it will return to zero at interval #0 for an amount of time corresponding to the PWM resolution (250ns).
An output can be set to a continuous "1" level by clearing the PWM and BRM values and setting POL = "1" (inverted polarity) in the PWM register. This allows a PWM/BRM channel to be used as an additional I/O pin if the DAC function is not required.

Table 7. 4-Bit BRM Added Pulse Intervals

(interval #0 not selected)

BRM 4-Bit Data	Incremental Pulse Interval #
0000	none
0001	i = 8
0010	i = 4,12
0100	i = 2,6,10,14
1000	i = 1,3,5,7,9,11,13,15

Figure 27. BRM pulse addition (PWM > 0)



DIGITAL TO ANALOG CONVERTER (Continued)

Figure 28. Simplified Filtered Voltage Output Schematic with BRM Added

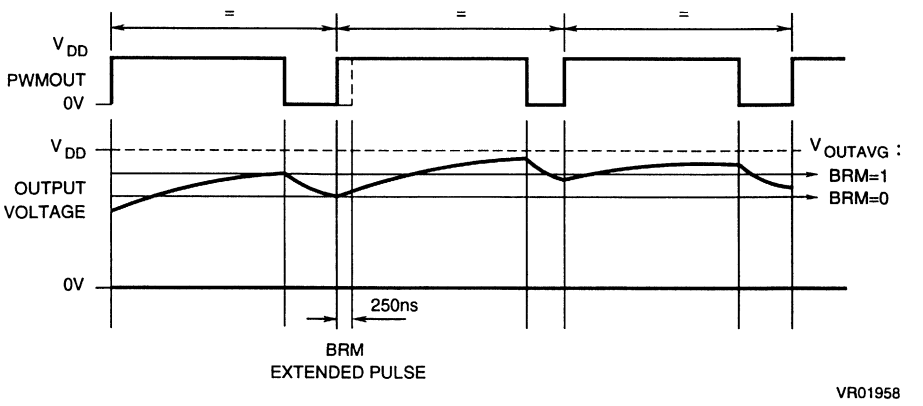
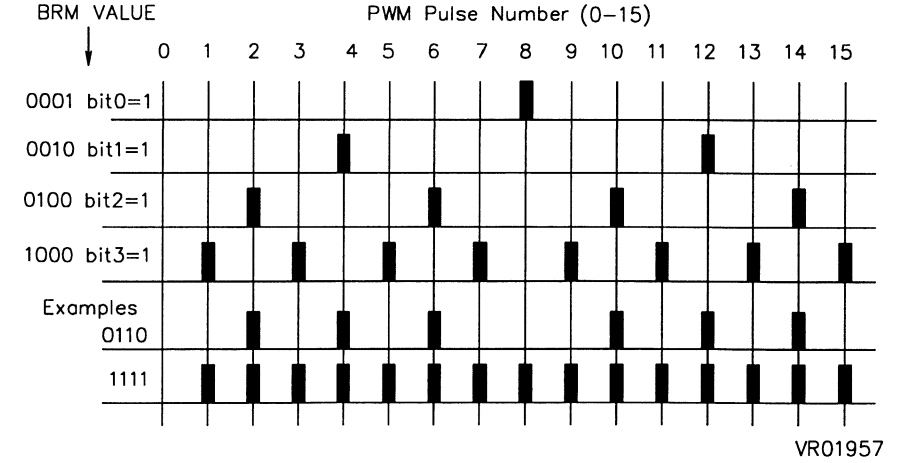


Figure 29. Graphical Representation of 4-Bit BRM added pulse Positions



DIGITAL TO ANALOG CONVERTER (Continued)

12-Bit PWM/BRM

The 12 Bits of the two channels of the 12-bit PWM/BRM generator are distributed as 6 PWM bits and 6 BRM bits. The two 12-bit channels correspond to PWM0 and PWM1, and outputs DA0 and DA1 respectively.

PWM Generation

The functionality of the PWM generation is equivalent to the PWM generation of the 10-bit PWM/BRM described in the previous paragraph and so will not be repeated here. Please refer to the previous paragraph for functionality, to be used in conjunction with the following Register description.

BRM Generation

A 6-bit BRM register defining the intervals where an incremental pulse (with duration of t_{CLK} = $1/f_{CLK}$=250ns) is added to the beginning of the original PWM pulse.

Table 8. 6-Bit BRM Added Pulse Intervals

(interval #0 not selected)

BRM 6-Bit Data	Incremental Pulse Interval #
000000	none
000001	i = 32
000010	i = 16, 48
000100	i = 8, 24, 40, 56
001000	i = 4,12, 20, 28, 36, 44, 52, 60
010000	i = 2,6,10,14,...,50, 54, 58, 62
100000	i = 1,3,5,7,9,..., 55, 59, 61, 63

2.5.3 Register Description

10-bit PWM/BRM REGISTERS

On a channel basis, the 10 bits are separated into two data registers:

A 6-bit PWM register corresponding to the binary weight of the PWM pulse.

A 4-bit BRM register defining the intervals where an incremental pulse is added to the beginning of the original PWM pulse. Two BRM channel values share the same register.

PULSE BINARY WEIGHT REGISTER

Register PWMi, i=2,17 (see register map)

Reset Value 1000 0000 (80h)

PWM Pulse Binary Weight

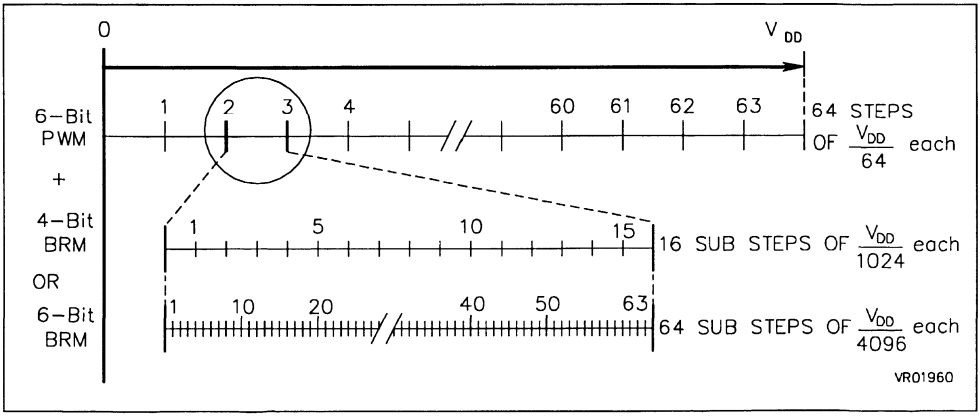
7							0
1	POL	P5	P4	P3	P2	P1	P0

b7 = **Reserved** (read as a "1")

b6 = **POL** *Polarity Bit.* When POL is set, output signal polarity is inversed; otherwise, no change occurs.

b5-0 = **P5-P0** *PWM Pulse Binary Weight* for channel i

Figure 30. Precision for PWM/BRM Tuning for V_{OUTEFF} (After Filtering)



DIGITAL TO ANALOG CONVERTER (Continued)

BRM REGISTER

Register BRMi and BRM(i+1), i=2,16

Reset Value: 0000 0000 (00h)

7							0
B7	B6	B5	B4	B3	B2	B1	B0

b7-4 = **B7-B4** *BRM Bits (channel i+1)*

b3-0 = **B3-B0** *BRM Bits (channel i)*

12-bit PWM/BRM REGISTERS

For each of the two channels, the 12 bits are separated into two data registers:

A 6-bit PWM register corresponding to the binary weight of the PWM pulse.

A 6-bit BRM register defining the intervals where incremental pulses are added to the beginning of the original PWM pulse.

PULSE BINARY WEIGHT REGISTER

Register PWMi, i=0,1 (see register map)

Reset Value: 1000 0000 (80h)

PWM Pulse Binary Weight

7							0
1	POL	P5	P4	P3	P2	P1	P0

b7 = **Reserved** (read as a "1")

b6 = **POL** *Polarity Bit.* When POL is set, output signal polarity is inversed; otherwise, no change occurs.

b5-0 = **P5-P0** *PWM Pulse Binary Weight* for channel i

BRM REGISTER

Register BRMi, i=0,1

Reset Value: 1100 0000 (C0h)

7							0
—	—	B5	B4	B3	B2	B1	B0

b7-6 = **Unused**

b5-0 = **B5-B0** *BRM Bits (channel i)*

Note : From the programmer's point of view, the PWM and BRM registers can be regarded as being combined to give one data value.

For example :

7	PWM						0		3	BRM	0
0	P	P	P	P	P	P	P	+	B	B	B	B

Effective* DAC value

11			8	7			=				0
0	P	P	P	P	P	P	P	B	B	B	B

* with external RC filtering

PWM/BRM OUTPUTS

The PWM/BRM outputs are assigned to the following pins (unless otherwise stated, they are 10-Bit PWM/BRM, push-pull/open-drain output (0-V_{DD}) configuration):

Table 9. PWM/BRM Pin Assignment

PWM/BRM	Pin	PWM/BRM	Pin
0(1,2)	DA.0	10(3)	DA.10
1(1)	DA.1	11(3)	DA.11
2(2)	DA.2	12(3)	DA.12
3	DA.3	13(3)	DA.13
4	DA.4	14(3)	DA.14
5	DA.5	15(3)	DA.15
6	DA.6	16(3)	DA.16
7	DA.7	17(3)	DA.17
8	DA.8		
9	DA.9		

Notes:

1. 12-Bit PWM/BRM
2. Fixed Open-Drain
3. Not present in 42 pin package

2.6 SERIAL PERIPHERAL INTERFACE

2.6.1 Introduction

The Serial Peripheral Interface (SPI) is an interface which allows several MCUs or peripherals to be interconnected within a single "black box" or on the same printed circuit board.

With the SPI interface, separate wires (signals) are required for data and clock. In the SPI format, the clock is not included in the data stream and uses a separate signal. An SPI system may be configured as containing one master MCU and several slave MCUs, or as a system in which a MCU is capable of being either a master or a slave.

The SPI features:

- Full duplex, three-wire synchronous transfers
- Master or slave operation
- 2 MHz (maximum) master bit frequency
- 4 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End of transmission interrupt flag
- Write collision flag protection
- Master-Master mode fault protection capability.

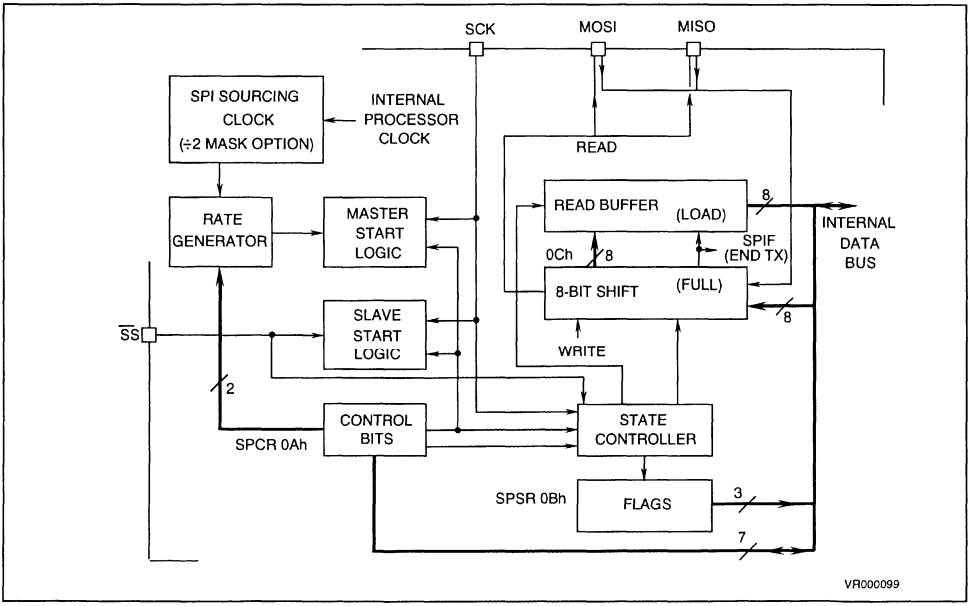
2.6.2 Functional Description

A block diagram of the serial peripheral interface (SPI) is shown in Figure 31. In a master configuration, the master start logic receives an input from the CPU (in the form of a write to the SPI rate generator data register) and originates the system clock (SCK) based on the internal processor clock. This clock is also used internally to control the control logic as well as the 8 bit shift register.

As a master device, data is parallel loaded into the 8 bit shift register (from the internal bus) during a write cycle and then shifted out serially to the MOSI pin for application to the slave device(s). During a read cycle, data is applied serially from a slave device via the MISO pin to the 8 bit shift register. After the 8 bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low (from a master device) at the $\overline{SS}$ pin and a system clock input (from the same master device) at the SCK pin. Thus, the slave is synchronized with the master. Data from the master is received serially at the slave MOSI pin and loads the 8 bit shift register.

Figure 31. Serial Peripheral Interface Block Diagram



Note. The user can select by mask option if the internal processor clock signals divided by 2 or not before entering the rate generator.

SERIAL PERIPHERAL INTERFACE (Continued)

After the 8 bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle.
During a write cycle, data is parallel loaded into the 8 bit shift register from the internal data bus and then shifted out serially to the MISO pin for application to the master device.

Figure 33 illustrates the MOSI, MISO and SCK master-slave interconnections. Note that the master $\overline{SS}$ pin is tied to a logic high and the slave $\overline{SS}$ pin is tied to a logic low.

2.6.3 Inputs and Outputs

The four basic signals (MOSI, MISO, SCK and $\overline{SS}$) are described in the following paragraphs. These signals are enabled for the corresponding pins of Port C by setting the SPIE bit of the Serial Peripheral control register (if these bits have been configured with a pull-up resistor through the PUPC bit of the Programmable Input/Output Register, then the pull-up is disabled and the port bits use the push-pull output configuration automatically). Each SPI signal function is described for both the master and slave mode.

Master Out Slave In (MOSI)
The MOSI pin is configured as a data output in the master (mode) device and as a data input in the slave (mode) device.

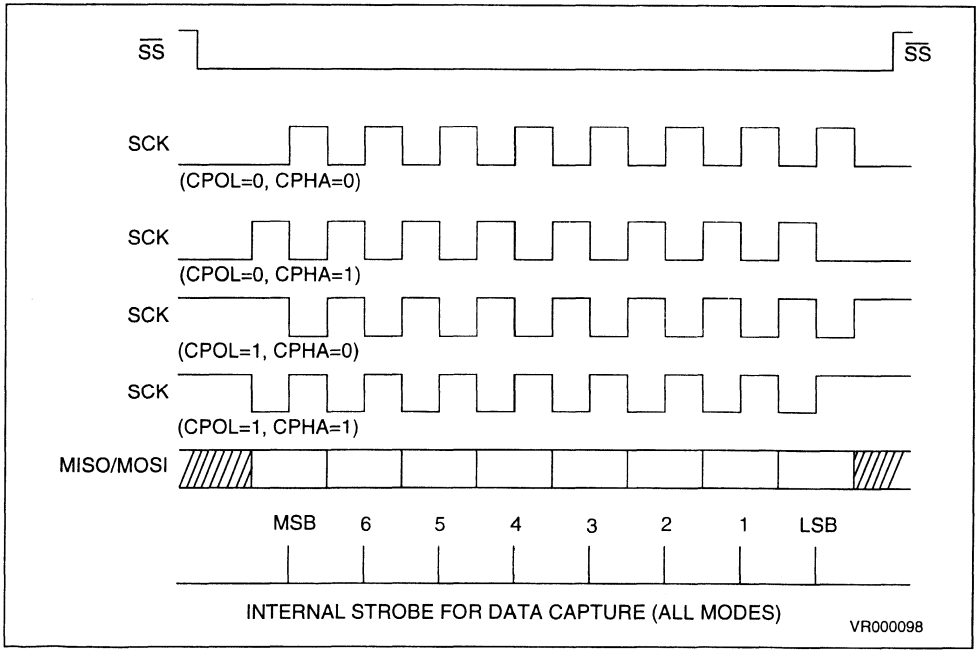
In this manner data is transferred serially from a master to a slave on this line, most significant bit first, least significant bit last. The timing diagram of Figure 32 summarizes the SPI timing diagram shown in the electrical specifications and shows the relationship between data and clock (SCK).

Figure 32 also shows the four possible timing relationships which may be chosen by using the SPI control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line one half-cycle before the clock edge (SCK) to allow the slave device to latch the data.

Both the slave device(s) and a master device must be programmed to similar modes for proper data transfer.

When the master device transmits data to a second (slave) device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in syn-

Figure 32. Data Clock Timing Diagram



SERIAL PERIPHERAL INTERFACE (Continued)

chronized with the same clock signal (which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A status bit (SPIF) in the serial peripheral status register (SPSR), location 0043h, is used to signify that the I/O operation is complete.

Configuration of the MOSI pin is a function of the MSTR bit in the serial peripheral control register (SPCR, location 0042h). When operating as a master, the user should set the MSTR bit to a logic one, giving the MOSI pin as an output.

Master In Slave Out (MISO)
The MISO pin is configured as data input in a master (mode) device and as data output in a slave (mode) device. Data is transferred serially from a slave to a master on this line, most significant bit first, least significant bit last. The MISO pin of a slave device is placed in the high-impedance state if it is not selected by the master, i.e., its $\overline{SS}$ pin is a logic one. The timing diagram of Figure 32 shows the relationship between data and clock (SCK).

In the master device, the MSTR control bit in the serial peripheral control register (SPCR, location 0042h) should be to a logic one (by the program) to allow the master device to receive data on its MISO pin. In the slave device, its MISO pin is enabled by the logic 1 level of the $\overline{SS}$ pin, i.e., if $\overline{SS}$ = 1, then the MISO pin is placed in the high-impedance state, whereas if $\overline{SS}$ = 0 the MISO pin is an output for the slave device.

Slave Select ($\overline{SS}$)
The slave select ($\overline{SS}$) pin is an alternate input function on PORT C pin 5 (PC5) enabled by setting the SPE bit of the serial peripheral control register. It receives an active low signal that is generated by a master device to enable slave devices to accept data.

To ensure that data will be accepted by a slave device, the $\overline{SS}$ signal line must be a logic low prior to occurence of SCK (system clock) and must remain low until after the last (eighth) SCK cycle.

Figure 32 illustrates the relationship between SCK and the data for two different level combinations of CPHA, when $\overline{SS}$ is pulled low. These are :

a) CPHA = 1 or 0, the first bit of data is applied to the MISO line for transfer, and,

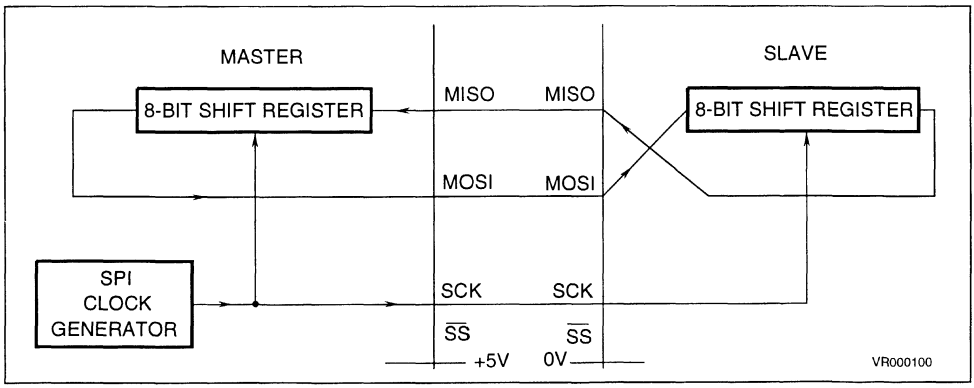
b) when CPHA = 0 the slave device is prevented from writing to its data register.

The state of the $\overline{SS}$ input and of the CPHA bit may have an effect on the I/O data register, refer to following section titled "Collision Detection".

A high level $\overline{SS}$ signal forces the MISO (master in/slave out) line to the high-impedance state. In addition, SCK and the MOSI (master out/slave in) line are ignored by a slave device when its $\overline{SS}$ signal is high.

When a device is a master, it constantly monitors its $\overline{SS}$ signal input for a logic low. The master device will become a slave device any time its $\overline{SS}$ signal input is detected low. This ensures that there is only one master controlling the $\overline{SS}$ line for a particular system.

Figure 33. SPI Master-Slave Interconnections



SERIAL PERIPHERAL INTERFACE (Continued)

When the $\overline{SS}$ line is detected low, the master clears the MSTR control bit (SPCR). Also, control bit SPE in the serial peripheral control register is cleared which causes the serial peripheral interface (SPI) to be disabled (port C SPI pins become inputs). The MODF flag bit in the serial peripheral status register (SPSR) is also set to indicate to the master device that another device is attempting to become a master. Two devices attempting to be master are normally the result of a software error, however, a system could be configured containing a default master which would automatically "takeover" and restart the system.

Serial Clock (SCK)

The serial clock is used to synchronize the movement of data both in and out of the device through its MOSI and MISO pins. The master and slave devices are capable of exchanging a data byte of information during a sequence of eight clock pulses. Since the SCK is generated by the master device, the SCK line becomes an input on all slave devices and synchronizes slave data transfer. The type of clock and its relationship to data are controlled by the CPOL and CPHA bits in the serial peripheral control register (SPCR) discussed below. Refer to Figure 32 for timing.

The master device generates the SCK through a circuit driven by the internal processor clock. A mask option allows the processor clock to be divided by two or not, allowing the input clock to be 4MHz or 2MHz (for 8MHz oscillator).
Two bits (SPR0 and SPR1) in the serial peripheral control register (SPCR) of the master device select the clock rate. The master device uses the SCK to latch incoming slave device data on the MISO line and shifts out data to the slave device on the MOSI line. Both master and slave devices must be operated in the same timing mode as controlled by the CPOL and CPHA bit in the serial peripheral control register. In the slave device, SPR0, SPR1 have no effect on the operation of the serial peripheral interface.

Collision Detection

An SPI collision is defined as when an attempt was made to write to the serial peripheral data register while a data transfer was taking place with an external device. The transfer continues uninterrupted; therefore, a write will be unsuccessful. A "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the operation of the MCU.

A collision of a write to the serial peripheral data register while an external data transfer is taking place can occur both in the master mode and the slave mode, although with proper programming the master device should have sufficient information to preclude this collision.

Collision in the master device is defined as a write of the serial peripheral data register while the internal rate clock (SCK) is in the process of transfer.

The signal on the $\overline{SS}$ pin is always high on the master device.

A collision in a slave device is defined in two separate modes. One problem arises in a slave device when the CPHA control bit is a logic zero.
When CPHA is logic zero, data is latched with the occurence of the first clock transition. The slave device has no way of knowing when that transition will occur; therefore, the slave device collision occurs when it attempts to write the serial peripheral data register after its $\overline{SS}$ pin has been pulled low. The $\overline{SS}$ pin of the slave device freezes the data in its serial peripheral data register and does not allow it to be altered if the CPHA bit is a logic zero. The master device must raise the $\overline{SS}$ pin of the slave device high between each byte it transfers to the slave device

The second collision mode is defined for the state of the CPHA control bit being a logic one. With the CPHA bit set, the slave device will receive a clock (SCK) edge prior to the latch of the first data transfer. This first clock edge will freeze the data in the slave device I/O register and allow the msb of the data onto the external MISO pin of the slave device. The $\overline{SS}$ pin low state enables the slave device but the drive onto the MISO pin does not take place until the first data transfer clock edge.
By definition of the second collision mode, a master device might hold a slave device $\overline{SS}$ pin low during a transfer of several bytes of data without a problem.

Detection of write collision is shown in the write collision status bit WCOL in the serial peripheral status register. The WCOL bit is only set if the I/O register is accessed while a transfer is taking place.

Unlike other SPI interfaces, there is no special case of collision undetected by the WCOL bits. The WCOL bit is totally reliable for collision detection.

Since the slave device operates asynchronously with the master device, the WCOL bit may be used as an indicator of a collision occurence. This also helps alleviate the user from a strict real-time programming effort.

SERIAL PERIPHERAL INTERFACE (Continued)

2.6.4 Register Description

SERIAL PERIPHERAL CONTROL REGISTER (SPCR) (0042h)

Reset condition: 00x0 xxxx

7							0
SPIE	SPE	Res	MSTR	CPOL	CPHA	SPR1	SPR0

b7 = **SPIE:** *Interrupt Enable* When the serial peripheral interrupts enable bit is high, it allows the occurence of a processor interrupt, and forces the proper vector to be loaded into the program counter if the serial peripheral status register flag bit (SPIF and/or MODF) is set to a logic one. It does not inhibit the setting of a status bit.
SPIE is cleared by reset.

b6 = **SPE:** *Output Enable Control* When the serial peripheral output enable control bit is set, all output drive is applied to the external pins and the system is enabled. When the SPE bit is set, it enables the SPI system by connecting it to the external pins thus allowing it to interface with the external SPI bus. The pins that are defined as output depend on which mode (master or slave) the device is in.
As the SPE bit is cleared by reset, the SPI system is not connected to the external pins upon reset.

b5 = **reserved**.

b4 = **MSTR** *Master/Slave Select* The master bit determines whether the device is a master or a slave. If the MSTR bit is a logic zero it indicates a slave device and a logic one denotes a master device. If the master mode is selected, the function of the SCK pin changes from an input to an output and the function of the MISO and MOSI pins are reversed. This allows the user to wire device pins MISO to MISO, and MOSI to MOSI, and SCK to SCK without incident.
The MSTR bit is cleared by reset, therefore, the device is always placed in the slave mode during reset.

b3 = **CPOL** *Clock Polarity.* The clock polarity bit controls the normal or steady state value of the clock when no data is being transferred. The CPOL bit affects both the master and slave modes. It must be used in conjunction with the clock phase control bit (CPHA) to produce the wanted clock-data relationship between a master and a slave device. When the CP0L bit is a logic zero, it produces a steady state low value at the SCK pin of the master device. If the CPOL bit is a logic one, a high value is produced at the SCK pin of the master device when data is not being transferred.
CPOL is not affected by reset.

b2 = **CPHA:** *Clock Phase* The clock phase bit controls the relationship between the data on the MISO and MOSI pins and the clock produced or received at the SCK pin. This control has effect in both the master or slave modes. It must be used in conjunction with the clock polarity control bit (CPOL) to produce the wanted clock-data relationship. In general the CPHA bit selects the clock edge which captures data and allows it to change states. It has its greatest impact on the first bit transmitted (MSB) in that it does or does not allow a clock transition before the first data capture edge.
CPHA is not affected by reset.

b1,b0 = **SPR1, SPR0:**
These two serial peripheral rate bits select one of four baud rates to be used as SCK if the device is a master. However, these 2 bits have no effect in the slave mode. The slave device is capable of shifting data in and out at a maximum rate which is equal to the CPU clock. A rate table is given below for the generation of the SCK from the master.
The SPR1 and SPR0 bits are not affected by reset.

SPR1	SPR0	Internal Processor Clock Divide[1]
0	0	2
0	1	4
1	0	16
1	1	32

Note 1. In this case, the SPI sourcing clock mask option (divider by 2) has not been selected.

SERIAL PERIPHERAL INTERFACE (Continued)

SERIAL PERIPHERAL STATUS REGISTER (SPSR) (0043h)

reset condition: 00x0 xxxx

7			6				0
SPIF	WCOL	–	MODF	–	–	–	–

The status flags which generate a serial peripheral interface (SPI) interrupt may be blocked by the SPIE control bit in the serial peripheral control register. The WCOL bit does not cause an interrupt. The serial peripheral status register bits are defined as follows :

b7 = **SPIF:***Serial Peripheral Data Transfer Flag*
This bit notifies the user that a data transfer between the device and an external device has been completed. With the completion of the data transfer, SPIF is set, and if SPIE is set, a serial peripheral interrupt (SPI) is generated. During the clock cycle SPIF is being set, a copy of the received data byte in the shift register is moved to a buffer. When the data register is read, it is the buffer that is read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not responded to the first SPIF, only the first byte sent is contained in the receiver buffer and all other bytes are lost.

The transfer of data is initiated by the master device writing to its serial peripheral data register.

Clearing the SPIF bit is accomplished by a software sequence of accessing the serial peripheral status register while SPIF is set and followed by a write to or a read of the serial peripheral data register.

While SPIF is set, all writes to the serial peripheral data register are inhibited until the serial peripheral status register is read. This occurs in the master device. In the slave device, SPIF can be cleared (using a similar sequence) during a second transmission; however, it must be cleared before the second SPIF in order to prevent an overrun condition. The SPIF bit is cleared by reset.

b6 = **WCOL:** *Write Collision Status.* If a "write collision" occurs, WCOL is set but no SPI interrupt is generated. The WCOL bit is a status flag only.

Clearing the WCOL bit is accomplished by a software sequence of accessing the serial peripheral status register while WCOL is set, followed by:

- A read of the serial peripheral data register prior to the SPIF bit being set, or
- A read or write of the serial peripheral data register after the SPIF bit is set.

A write to the serial peripheral data register (SPDR) prior to the SPIF bit being set, will result in generation of another WCOL status flag. Both the SPIF and WCOL bits will be cleared in the same sequence. If a second transfer has started while trying to clear (the previously set) SPIF and WCOL bits with a clearing sequence containing a write to the serial peripheral data register, only the SPIF bit will be cleared.
WCOL is cleared by reset.

b4 = **MODF:**
The function of the mode fault flag is defined for the master mode (device). If the device is a slave device the MODF bit will be prevented from toggling from a logic zero to a logic one; however, this does not prevent the device from being in the slave mode with the MODF bit set.

The MODF bit is normally a logic zero and is set only when the master device has its $\overline{SS}$ pin pulled low. Toggling the MODF bit to a logic one affects the internal serial peripheral interface (SPI) system in the following ways:

- MODF is set and SPI interrupt is generated if SPIE = 1.
- The SPE bit is forced to a logic zero. This blocks all output drive from the device and, disables the SPI system.
- The MSTR bit is forced to a logic zero, thus forcing the device into the slave mode.

Clearing the MODF is accomplished by a software sequence of accessing the serial peripheral status register while MODF is set followed by a write to the serial peripheral control register.

To avoid any multi slave conflict in the case of a system of several MCUs, the $\overline{SS}$ pin must be pulled high during the clearing sequence of MODF. Control bits SPE and MSTR may be restored to their original set state during this clearing sequence or after the MODF bit has been cleared. Hardware does not allow the user to set the SPE and MSTR bit while MODF is a logic one unless it is during the proper clearing sequence. The MODF flag bit indicates that there might have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state. The MODF bit is cleared by reset.

SERIAL PERIPHERAL INTERFACE (Continued)

SERIAL PERIPHERAL DATA I/O REGISTER (SPDR) (0040h)
reset condition: undefined

7							0
–	–	–	–	–	–	–	–

The serial peripheral data I/O is used to transmit and receive data on the serial bus. Only a write to this register will initiate transmission/reception of another byte and this will only occur in the master device. A slave device writing to its data I/O register will not initiate a transmission. At the completion of transmitting a byte of data, the SPIF status bit is set in both the master and slave devices.

A write or read of the serial peripheral data I/O register, after accessing the serial peripheral status register with SPIF set, will clear SPIF.

During the clock cycle the SPIF bit is being set, a copy of the received data byte in the shift register is being moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not internally responded to clear the first SPIF, only the first byte is contained in the receive buffer of the slave device; all others are lost. The user may read the buffer at any time. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated or an overrun condition will exist.

A write to the serial peripheral data I/O register is not buffered and places data directly into the shift register for transmission.

The ability to access the serial peripheral data I/O register is limited when a transmission is taking place. It is important to read the discussion defining the Write Collision and SPIF status bits to understand the utilization limits of the serial peripheral data I/O register.

2.6.5 Single Master and Multi Master Configurations

There are two types of SPI systems, single master system and multi-master systems.

A typical single master system may be configured, using a MCU as the master and fours MCUs as slaves. The MOSI, MISO and SCK pins are all wired to equivalent pins on each of the five devices. The master device generates the SCK clock,the slave devices all receive it. Since the MCU master device is the bus master, it internally controls the function of its MOSI and MISO lines, thus writing data to the slave devices on the MOSI and reading data from the slave devices on the MISO lines.

The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices. A slave device is selected when the master device pulls its $\overline{SS}$ pin low. The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices. Note that the slave devices do not have to be enabled in a mutually exclusive fashion except to prevent bus contention on the MISO lines. For example, three slave devices, enabled for a transfer, are permissible if only one has the capability of being read by the master. An example of this is a write to several display drivers to clear a display with a single I/O operation.

To ensure that proper data transmission is occuring between the master device and a slave device, the master device may have the slave device respond with a previously received data byte (this data byte could be inverted or at least be a byte that is different from the last one sent by the master device). The master device will always receive the previous byte back from the slave device if all MISO and MOSI lines are connected and the slave has not written its data I/O register. Other transmission security methods might be defined using ports for handshake lines or data bytes with command fields.

A multi-master system may also be configured by the user. An exchange of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system. The major device control that plays a part in this system is the MSTR bit in the serial peripheral control register and the MODF bit in the serial peripheral status register.

2.7 EAST-WEST PIN CUSHION CORRECTION (EWPCC) GENERATOR

2.7.1 Introduction

The function of the East-West Pin Cushion Correction generator (EWPCC) is to correct the horizontal synchronisation timing signals to compensate for the pin-cushion effect on the display screen, normally caused by the magnetic display components.

The EWPCC operates by storing, during the fabrication process of the monitor, 256 coefficients related to the pin cushion effect in a dedicated memory (EEPROM) and then allowing real-time correction by sending the stored coefficients out through the dedicated D/A Converter (DAC). The analog output is added (off-chip) to the deflection voltage that determines the starting point for the next line on the screen.
These coefficients are uniformly distributed over the effective vertical refresh field of the screen.

Figure 34. Pin-cushion Effect

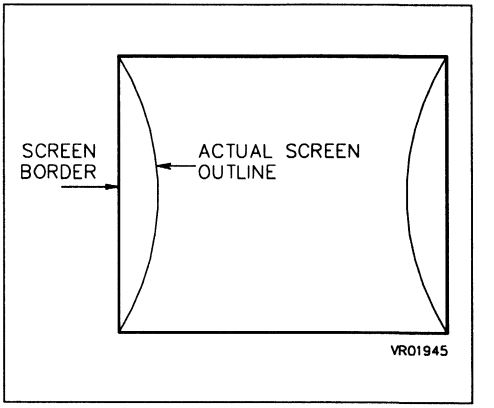
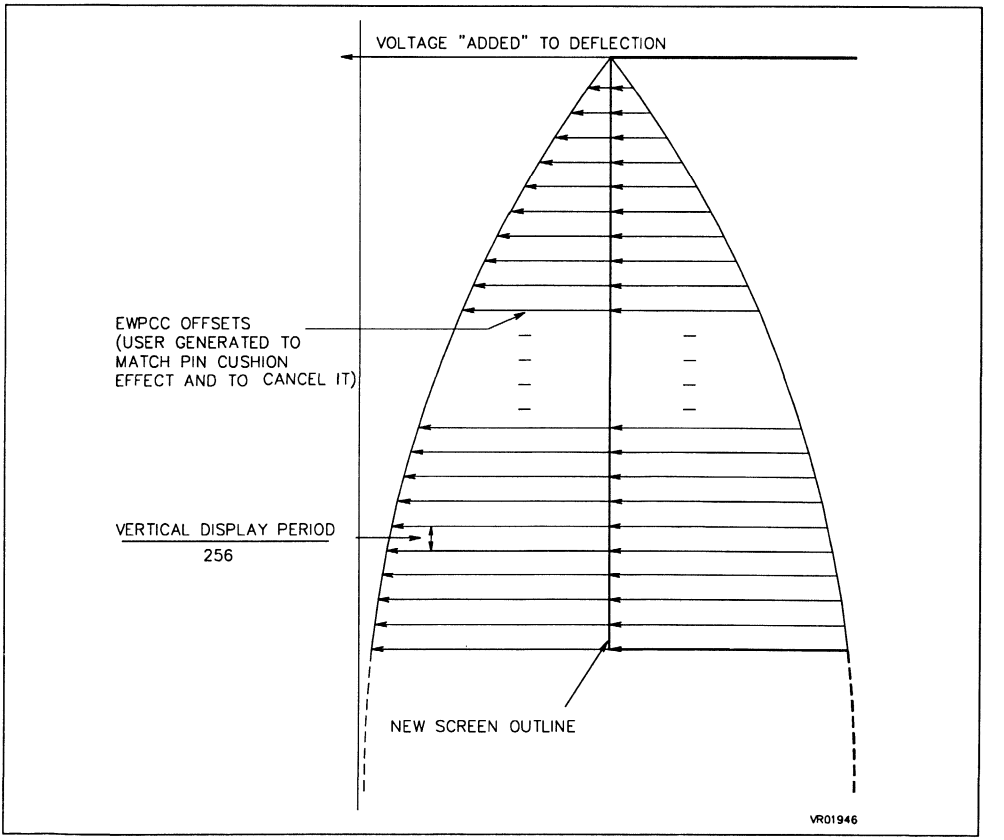


Figure 35. EWPC Coefficients Working on Pin-cushion



EWPCC GENERATOR (Continued)

2.7.2 Functional Description

The EWPCC output is based on the signal present on the VFBACK input pin. The input signal is required to be asserted during the Vertical Fly-Back period (with a fixed positive polarity assumed).
Warning: when the EWPCC is used, this I/O pin must be set to input mode; if the EWPCC is not used, the pin is available for normal output functions.

For correct operation the frequency of the internal clock of the ST7271 must be 1024 times greater than the flyback frequency.

In normal operation (after initialization), the EWPCC EEPROM is sequentially scanned by the address generator during the time interval that VFback is negated, that is, during the active vertical display period. When VFback is asserted (fly-back period) the address generator is reset, so that the address points to the first EEPROM position and the DAC register is loaded with the first coefficient.

The address generator consists of a 17-bit processor (8-bit up-counter + 9-bit up/down-counter), data/control registers and logic to minimize rounding-up effect.

Figure 37. VFBACK Input Timing

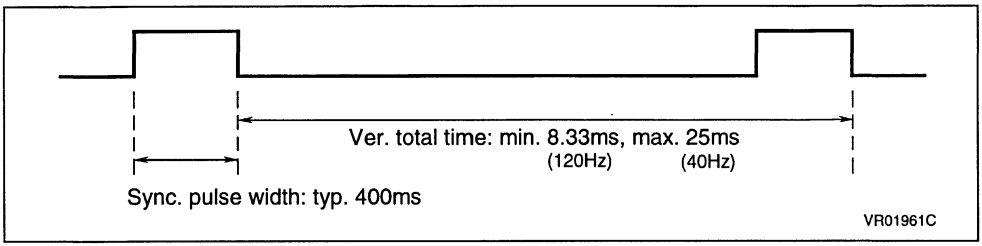
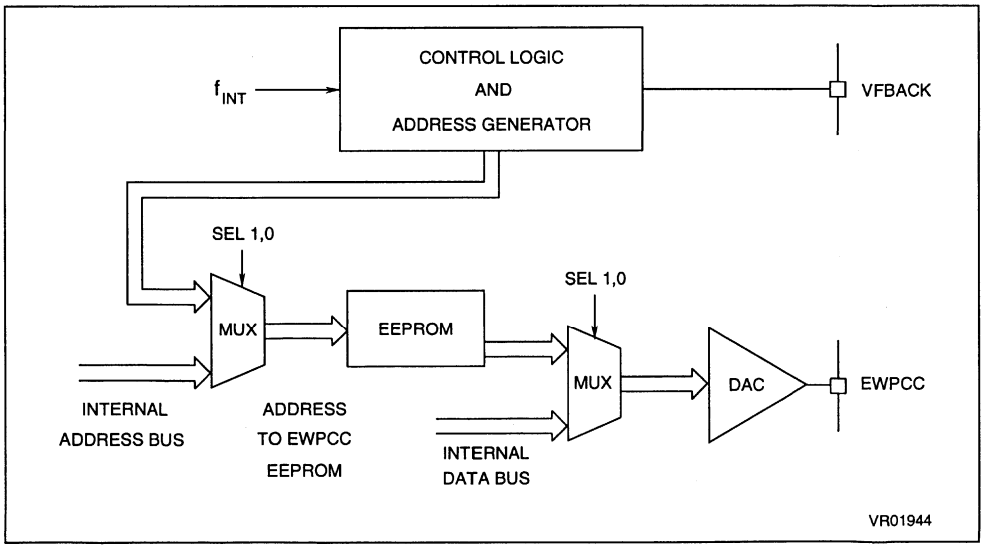


Figure 36. EWPCC Block Diagram



EWPCC GENERATOR (Continued)

EWPCC Operation

When the EWPCC function is disabled (SEL1,0 = [0,0]), the CPU may fill the EEPROM memory with the EWPCC coefficients calculated or measured during production test or transferred from a preset table within memory.

In normal operation, when enabled, 256 analog levels on the EWPCC output (DAC analog output) are generated in evenly-spaced time/space intervals during the time that VFBACK is negated.
To do so, the EWPCC performs two operating modes that are automatically chained once the sequence is triggered by writing a "1" to the control bit INI in the EWPCC1 register:

a) Acquisition Mode:

The output of the 8-bit prescaler, fed by the CPU clock (f_{CLK}), is counted up by a 9-bit counter during negation of VFback. The rising edge of VFback captures the 9-bit counter value and the 8-bit prescaler MSB into the data registers, as follows:

8-Bit Prescaler MSB -> EWPCC0.0

9-Bit Counter 7 LSB's -> EWPCC0.7,1

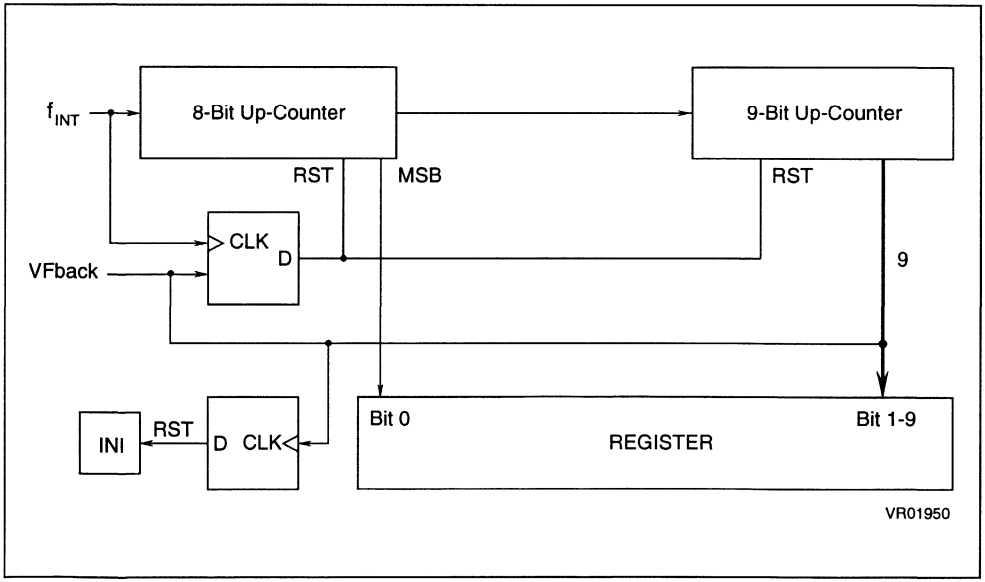
9-Bit Counter 2 MSB's -> EWPCC1.1,0

Thus the 9-bit counter contains a value proportional to the duration that VFback is low. As the lower 7 bits of the prescaler are not passed through to the next stage, there is an effective division by 256, so the value in the counter can be regarded as the number of f_{INT} cycles to give the time for the complete non-retrace period equally divided by 256.

The counter and the INI control bit (EWPCC1,2) are then automatically reset and the hardware is switched to the address generation mode.

In order to minimize the truncation effect on the captured value accuracy, the overall 17-bit counter (8-bit prescaler + 9-bit counter) is preset to 00040h during assertion of VFBACK.

Figure 38. Acquisition Mode Block Diagram



EWPCC GENERATOR (Continued)

Figure 39. Acquisition Mode Timing

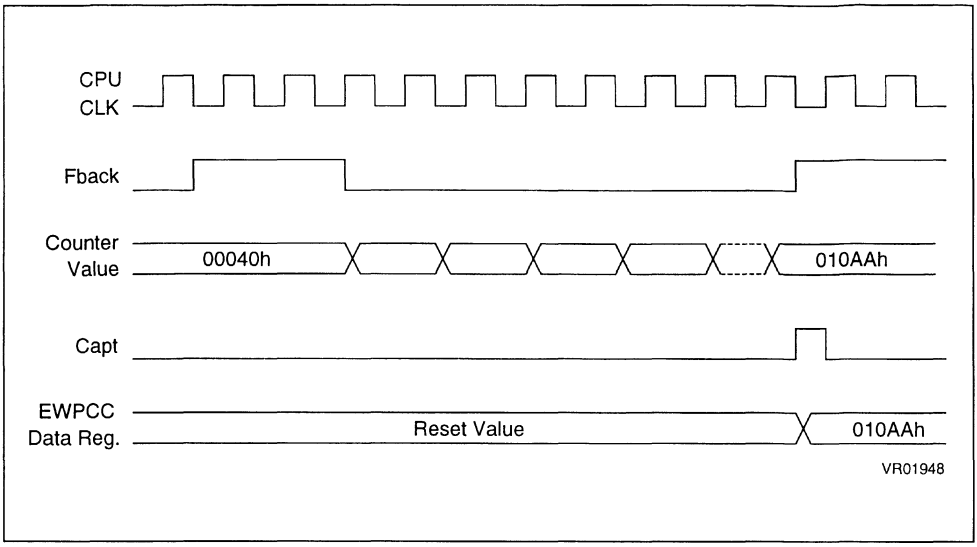
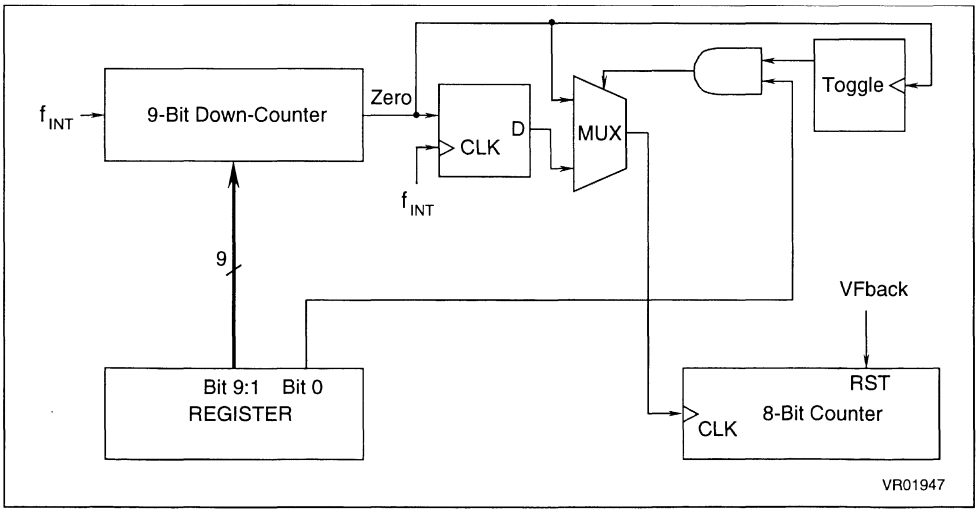


Figure 40. Address Generation Mode Block Diagram



EWPCC GENERATOR (Continued)

Figure 41. Address Clock Generation (Division by N)

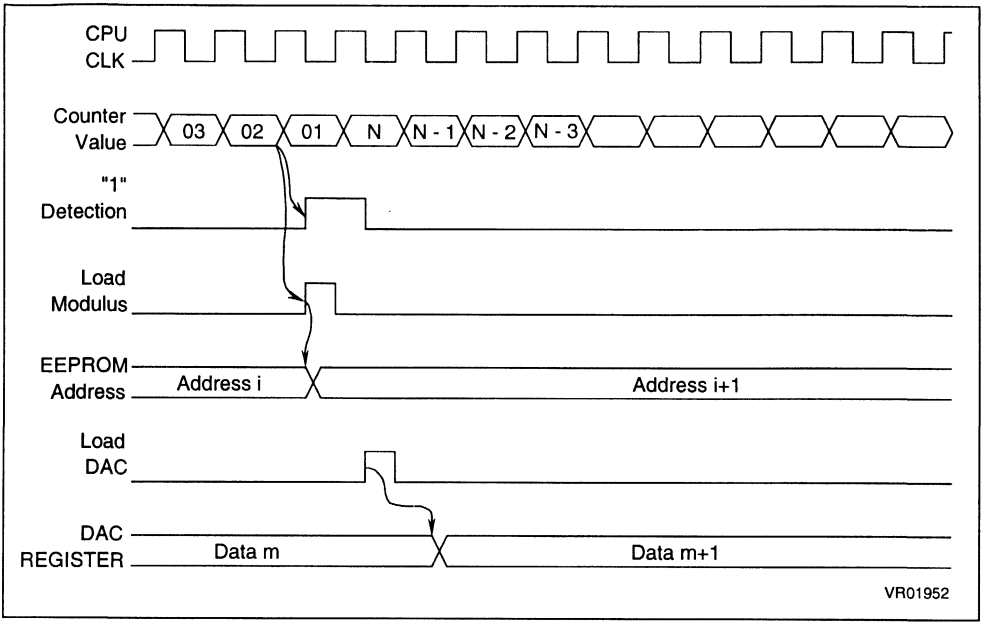
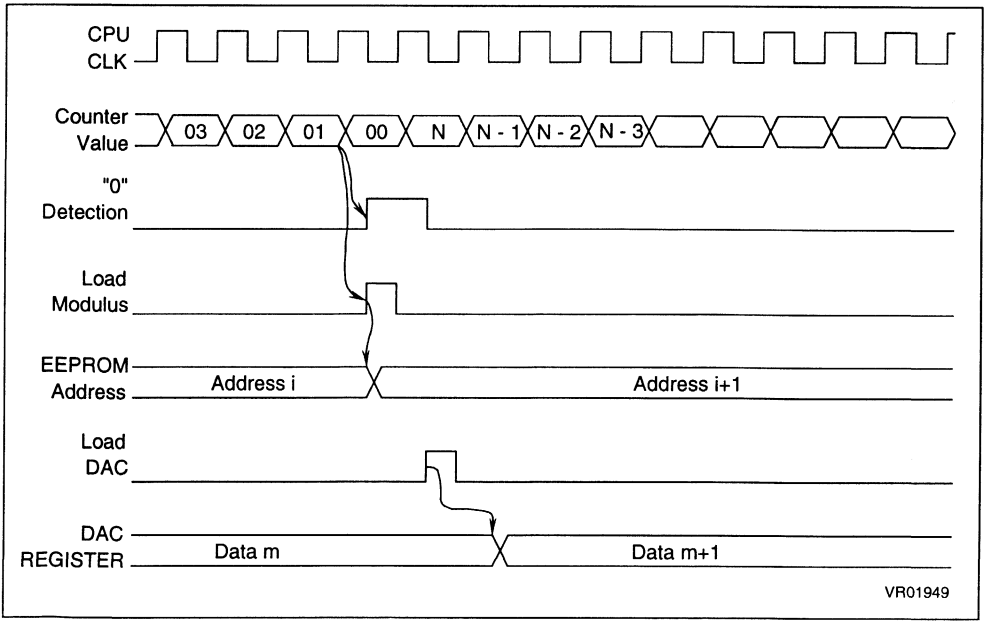


Figure 42. Address Clock Generation (Division by N+1)



EWPCC GENERATOR (Continued)

b) Address Generation Mode:

Once the acquisition pass is done, addresses to the EEPROM are continuously generated. The 17-bit processor is reconfigured as:

- A 9-bit modulus (divide-by-N/N+1) down-counter, clocked by the CPU clock (f_{CLK}), and reloaded on zero detection with the value captured in the acquisition mode.
- An 8-bit up-counter, clocked by zero detection in the 9-bit counter, which generates the EEPROM addresses.

The division ratio N corresponds to the 9 MSB's of the value captured in the acquisition mode (EWPCC1,0 + EWPCC07,1) which is equivalent to the original number of f_{INT} cycles to achieve the display time of VFBACK low divided by 256. The effect of this is then to divide the VFBACK time evenly by 256 and to increment the address to the EEPROM at each interval.

The choice of N or N+1 is controlled by EWPCC0.0. If this bit is "0", the counter modulus is always N, otherwise an extra step is introduced into the counter to give N+1 and the counter divides by N and N+1 alternately. This algorithm performs a division by (N + 0.5).

Note. After reset, the EWPCC function is disabled and the INI bit is cleared. By writing SEL1,0 = [1,1] and INI=1, the EWPCC function is enabled and the acquisition mode starts.

2.7.3 EWPCC Operation Modes

The EEPROM content addressed by the address counter is loaded into the Digital to Analog converter (DAC) 8-bit data register through a multiplexer and then sent to the D/A Converter. The other input to the multiplexer is the internal Data Bus.

This architecture allows the user the alternative of different configurations, selected by the control bits SEL1, 0.

- SEL1, 0 = [1,1]:

EWPCC Function Enabled (Normal Operation): EEPROM is addressed by the EWPCC Address Sequencer and its values directly sent to the DAC register through a dedicated bus.

- SEL1,0 = [0,1]:

EWPCC Function Disabled and Direct Loading (by a Single Instruction) to the DAC Register from the EEPROM through the dedicated bus. This direct transfer is synchronized with a dummy read instruction of the EEPROM with the addresses generated by the CPU. In this dummy operation, EEPROM data is not loaded onto the main data bus, but can be accessed by reading the DAC register.

- SEL1,0 = [0,0]

EWPCC Function Disabled. EEPROM and DAC in Stand-Alone Mode.

Whenever the dedicated path between the EEPROM and the Digital to Analog Converter is enabled, after accessing the EEPROM, the byte read is sent to the DAC data register. If the DAC is configured as stand-alone, its data register is directly loaded by the CPU through the internal data bus.

Tied to V_{DDA} (Analog VDD, 8V ± 8%) and V_{SSA} (Analog Ground), the DAC is monotonically crescent as it is constructed by a resistive ladder with 256 levels (V_{STEP} = 15.625mV) - between 2V (00h) and 5.9843V (FFh) - output to an unit-gain amplifier and then output to the dedicated pin (EWPCC).

The simulated electrical characteristics are listed in the table below, for a 15pF load capacitance and a 0-70°C temperature range.

DAC Electrical Characteristics

Variable	Min	Typ	Max
Output Current (µA)	340	1050	2200
Response Time (µs) (1)	3.0	3.6	8.0

Note 1. From the moment the new value is loaded onto the DAC register to 90% of the output steady value, for a 2V-step variation.

The architecture of the EWPCC also gives the user the possibility (if necessary) to adjust by software the captured value on the generation mode by writing into the EWPCC registers the new modulus that will be transferred to the counter upon the following underflow condition.

The EWPCC function may also be used for the generation of other waveforms by suitable programming of the EWPCC EEPROM.

Tolerance

The truncation error on the acquired value corresponds to -0.25 ≤ Error ≤ 0.25 (with the algorithm used) which can be related to a maximum shift of 32µm on the vertical axis of the screen, at the 256th address (or a maximum 16µm at the 128th) with respect to the ideal case of zero truncation error when generating evenly-spaced intervals by dividing the time interval in which VFBACK is negated by 256.

EWPCC GENERATOR (Continued)

2.7.4 Register Description

EWPCC0 REGISTER (000Dh)

Reset Value: 0000 0000 (00h)

7							0
M7	M6	M5	M4	M3	M2	M1	M0

b7-1 = **M7-M1:** *9-Bit Counter 7 LSBs.* These bits correspond to the 7 least-significant bits of the captured value from the 9-bit counter when in acquisition mode.

b0 = **M0:** *8-Bit Prescaler MSB.* This bit corresponds to the most significant bit of the 8-bit prescaler, captured during acquisition mode.

EWPCC1 REGISTER (000Eh)

Reset Value: 1100 0000 (C0h)

7							0
1	1	0	SEL1	SEL0	INI	M1	M0

b7-6 = **Reserved** (read as "1")

b5 = **Reserved** *MUST BE PROGRAMMED AS "0"*

b4-3 = **SEL1,SEL0:** *Mode Selection Bits.* These read/write bits allow selection of configuration, as follows:

SEL1	SEL0	
0	0	EWPCC Function Disabled EEPROM and DAC Stand Alone
0	1	EWPCC Function Disabled EEPROM addressed by CPU and its data sent directly to DAC
1	0	Reserved
1	1	EWPCC Function Enabled EEPROM addressed by EWPCC block Data sent to DAC Register

b2 = **INI:** *Acquisition Mode Initialization/Status Bit.* Writing a "1" to INI triggers the acquisition mode, after detection of VFback rising edge. During this mode, INI is read as a "1". Once acquisition is completed, this bit is cleared by hardware and address generation mode is chained.

b1-0 = **M1-M0:** *9-bit Counter Value MSB's.* These bits correspond to the two most-significant bits of the captured value from the 9-bit counter on acquisition mode.

Notes.

When EWPCC mode is selected (SEL1,0 = [1,1]), any writing operation into the EEPROM control register is blocked by hardware.

- When SEL0 = "1", the EEPROM value is not transferred onto the databus.
- During acquisition mode, re-writing the INI control/status bit as "1" will not have any effect - the acquisition procedure already in progress will not be reset.

DAC REGISTER (003Eh)

Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

b7-0 = **D7-D0:** *DAC Input Binary Byte.* This byte corresponds to the binary value to be converted onto an analog signal.

EWPCC EEPROM CONTROL REGISTER (000Fh)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
Res	Res	Res	—	—	E2ERA	E2LAT	E2PGM

This register contains the bits required to read, erase and program the EWPCC EEPROM. They are defined as follow:

b7-5 = **Reserved**, must be held to "0"

b4,3 = **Unused**

b2 = **E2ERA:** *EEPROM Erase.* E2ERA must be set to "1" for an erase operation. It must be set after or at the same time as E2LAT. It cannot be changed once an EEPROM address is selected. It is held low when E2LAT is low. It is therefore automatically reset when E2LAT is reset.

b1 = **E2LAT:** *EEPROM Latch Enable.* When E2LAT is reset to "0", data can be read from the EEPROM. When it is set to "1" and E2PGM reset to "0", a write into the EEPROM array causes the data to be latched, according to the address into one of 8 data registers. An additional internal flag is latched to select the row. The selected columns and row determine the locations involved in the next erase or programming operation. E2LAT must be cleared after each programming or erase operation. E2ERA and E2PGM are forced low when E2LAT is low.

b0 = **E2PGM:** *EEPROM Program Mode.* This bit allows the internal charge pump to be switched on or off. When set to "1", the charge pump generator is on and the high voltage is applied to the EEPROM array. When low, the charge pump generator is off. E2PGM can only be reset by resetting E2LAT.

2.8 A/D CONVERTER

The Analog to Digital converter is a single 8-bit successive approximation converter. Analog voltages from external sources are input to the converter through up to 8 pins (in the 42-pin DIL package, the selection is made from only 4 pins). The result of the conversion is stored in the 8-bit Result Data Register. The A/D converter is controlled through the A/D Converter Control/Status Register.

2.8.1 Functional Description

The A/D converter is switched on by setting the A/D Converter ON bit (ADON) of the A/D Converter Control/Status Register. A delay time is then required for the converter to stabilize (see characterization section).

When the A/D function is on, pins PB0/AIN0-PB7/AIN7 (pins PB0/AIN0- PB3/AIN3 in the 42-pin package) can be used as analog inputs. The inputs must first be enabled for analog input by setting the corresponding bit(s) of the Port B Configuration Register as described in the Section on I/O Ports. Bits CH3-CH0 of the A/D Converter Control/Status Register (at address 0009h) then select the channel to be converted. The high and low level reference voltages are connected to pins V_{DD} and V_{SS}.

When switched on, the A/D converter performs a continuous conversion of the selected channel. When a conversion is completed (16 µs for f_{INT} = 4 MHz), the results is loaded into the read only Result Data Register (at address 0008h) and the COCO (Conversion Complete) flag is set. No interrupt is generated. Any write to the A/D Converter Control/Status Register aborts the current conversion, resets the COCO flag and starts a new conversion.

The A/D converter is ratiometric. An input voltage equal or greater than V_{DD} converts to FFh (full scale) without overflow indication if greater. An input voltage equal to V_{SS} converts to 00h. The conversion is monotonic: the results never decrease if the analog input does not and never increase if the analog input does not.

Using a pin or pins as analog inputs does not affect the ability to read port B as logic inputs.

The 8-bit conversion is accurate to within ± 2 LSB. The minimal conversion time is 16ms (A/D clock frequency at 2 MHz). The A/D converter clock is generated from the CPU clock divided by 2.

The A/D converter can be switched off by resetting the ADON bit. This feature allows the reduction of power consumption when no conversion is in progress. The A/D converter is disabled after power-on and external resets.

The A/D converter is not affected by WAIT mode but, in power sensitive applications, it can be switched off before entering this mode. When the MCU enters the STOP mode with the A/D converter enabled, the A/D clocks are stopped and the converter is disabled until the STOP mode is exited and the startup delay has elapsed. A stabilization time is also required before accurate conversions can be performed.

The reference supply of the converter uses the power supply lines V_{DD} and V_{SS}. The accuracy of the conversion may therefore be degraded by voltage drops in the heavily loaded power lines.

A/D CONVERTER (Continued)

2.8.2 Register Description

A/D CONTROL/STATUS REGISTER (0009h)

Reset condition: 0000 0000 (00h)

7							0
COCO	0	ADON	—	0	CH2	CH1	CH0

b7 = **COCO:** *Conversion Complete.* COCO is set as soon as a new conversion can be read from the Result Data Register. COCO is cleared by reading the result or writing to the A/D Converter Control/Status Register.

b6 = **Reserved,** must be programmed to 0

b5 = **ADON:** *A/D converter On.* ADON allows the A/D converter to be switched on and off in order to reduce consumption when needed. When turned on (ADON = 1), a delay time is necessary for the current to stabilize. Conversions can be inaccurate during this time.

b4 = **Not used**

b3 = **Reserved,** must be programmed to 0

b2-b0 = **CH2-CH0:** *Channel Selection* These bits select the analog input to convert .

CH2-CH0	Pin	Channel
0 0 0	PB0	AIN0
0 0 1	PB1	AIN1
0 1 0	PB2	AIN2
0 1 1	PB3	AIN3
1 0 0	PB4	AIN4
1 0 1	PB5	AIN5
1 1 0	PB6	AIN6
1 1 1	PB7	AIN7

A/D DATA REGISTER (0008h)

reset condition: undefined

7							0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

b7-b0 = **AD7-AD0:** *Analog Converted Value.*

NOTES

SOFTWARE AND CHARACTERISTICS

3.1SOFTWARE DESCRIPTION

3.1.1 Instruction Set

The ST7 instruction set is an 8 bit industry standard instruction set that can be divided into five major groups. All instructions of each group have the same addressing modes.
Refer to ST7 MACRO ASSEMBLER USER'S GUIDE and ST7 PROGRAMMING MANUAL for detailed information.

Group 1 : Register/Memory And Absolute Jump Group In this group most instructions contain two operands. One operand is inherently defined as either the accumulator or an index register. The other operand is fetched from memory using one of the allowed addressing modes. The absolute jump instructions are included in this group because they can use most of the addressing modes of the register/memory instructions.

Examples: LD <ea>, a. This means that the memory byte located at address <ea> is loaded with the 8-bit content of the accumulator A.

The list of the instructions of this group is given in Table 10.

Group 2 : Read - Modify - Write Group These instructions read a register or a memory location, modify its content and write the new value back.

Example : RRC <ea>. This means that the content of the memory byte located at address <ea> is rotated right through the carry bit, the result is written in the memory <ea> and the carry bit.

The list of the instructions of this group is given in Table 11.

Group 3 : Bit Manipulation And Test Group Bit manipulation instructions can set or clear any bit within the first 256 memory locations, except for ROM (020h - 04F) and read-only registers located at addresses 03h, 0Bh, 010h, 013h, 014h and 015h.

Example: BSET <ea>, #b. This sets the bit #b of memory location <ea>.

Test instructions can test any bit of the first 256 memory locations and jump conditional within an 8 bit PC-relative displacement.

Example: BTJT <ea>, #b, ee. This corresponds to the relative jump (displacement = ee) if bit number #b of memory location <ea> is set. (Bit test and jump if true).

The list of the instructions of this group is given in Table 12.

Group 4 : PC-Relative Jump Group These instructions execute a PC-relative jump (8-bit displacement) depending on the state of the flag bits inside the condition code register (H, I, N, Z, C flags).

Example: JRC ee. This means jump with displacement ee from actual the PC value if the carry bit is set, else execute the next instruction.

The list of the instructions of this group is given in Table 13.

Group 5 : Miscellaneous Group These instructions are mainly control instructions on registers, stack, interrupts, subroutines and power down modes.

The multiply instruction is included in this group. It performs an 8 x 8 bit unsigned multiplication between one index register and the accumulator. The result is given as 16 bits, with the high order byte in the index register and the low order one in the accumulator.

The list of the instructions of this group is given in Table 14.

3.1.2 Addressing Modes

The CPU uses 9 main addressing modes to provide the programmer with an opportunity to optimize his code in most applications.

The various indexed addressing modes make it possible to locate data labels, code conversion tables and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) allow access of tables throughout memory.

Short absolute (direct) and long absolute (extended) addressing modes are also included. Extended addressing permits jump instructions to reach all memory.

The various addressing modes differ from each other in computing the effective address (EA) i.e. in calculating the address to or from which the argument of an instruction is fetched or stored. The LSBEA is the least significant byte of the EA; the MSBEA is its most significant byte. The 17 addressing modes of the processor are described below.

The table of symbols is given in Table 9 while the effective address coding is given in Table 8.

SOFTWARE DESCRIPTION (Continued)

In order to extend the number of op-codes available for an eight bit CPU (256 op-codes), three "pre-byte" op-codes have been defined. These pre-byte have to be seen as pre-instructions that modify the meaning of the instruction they precede. The whole instruction becomes:

PC-1 End of previous instruction

PC Pre-byte

PC + 1 Op-code

PC + 2 Additional word (0 to 2) according to the number of byte required to compute the effective address.

The pre-bytes enable instructions in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or of the instruction using a direct addressing mode.

The pre-bytes are :

PDY : 90h Transform an instruction in X using immediate, direct, indexed, direct bit or inherent addressing modes to an instruction in Y using the same addressing mode.

PIY : 91h Transform an instruction using X indexed addressing mode to an instruction using indirect Y indexed addressing mode.

PIX : 92h Transform an instruction using direct, direct bit or direct relative addressing mode to an instruction using the corresponding indirect addressing mode. It also transforms an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.

The pre-byte is completely user transparent. It is part of the assembly code.

Table 8. Source Coding

Addressing Mode	Source Coding	Example
Immediate	#nn	LD a, #0Ah
Direct	ad8	LD a, 0Ah
Extended	ad16	LD a, 10EAh
Indexed no offset	(iX)	LD a, (X)
Indexed 8 bit offset	(d8, iX)	LD a, (1Bh, Y)
Indexed 16 bit offset	(d16, iX)	LD a, (100Ah, X)
Memory indirect short	[ad8]	LD a, [1Bh]
Memory indirect long	[ad16]	LD a, [100Ah]
Memory indirect short indexed	([ad8], iX)	LD a, ([1Bh], X)
Memory indirect long indexed	([ad16], iX)	LD a, ([100Ah], Y)

Table 9. Table of Symbols used

a	Accumulator	nn	8 bit immediate value
iX	Index register (either X o r Y)	ad8	8 bit address
X	X index register	ad16	16 bit address
Y	Y index register	d8	8 bit signed offset
S	Stack pointer	d16	16 bit signed offset
CC	Condition code register	ee	8 bit PC relative displacement
<ea>	Effective address	b	3 bit number

SOFTWARE DESCRIPTION (Continued)

The addressing modes are discussed in the following paragraphs.

Inherent. In inherent instructions, there is no EA as there is no operand to fetch or store. All the information needed to execute the instruction is contained in the op-code. Operations specifying only an index register or the accumulator, and no other arguments, are included in this mode.

Immediate. In immediate addressing the operand is stored in the byte immediately following the op-code.

Direct Modes

Direct. In the direct addressing mode, the effective address is contained in a single byte following the op-code byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two byte instruction.

Extended. In the extended addressing mode, the effective address of the argument is contained in the two bytes following the op-code. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory.

Indexed Modes

Indexed Without Offset. In the indexed without offset addressing mode, the effective address is contained in one index register (X or Y). This addressing mode can therefore access the first 256 memory locations. These instructions are only one byte long.

This mode is mainly used to move a pointer through a table or to address a frequently referenced RAM or I/0 location.

Indexed 8 Bit Offset. The EA is obtained by adding the 8-bit unsigned contents of the second instruction byte to the 8-bit unsigned content of the appropriate index register. This mode allows addressing of 256 locations of the 511 lowest memory locations.

Indexed 16 Bit Offset. The EA is obtained by adding the 16-bit unsigned value composed of the second (MSB) and the third (LSB) instruction bytes to the 8-bit unsigned content of the appropriate index register. This mode allows addressing of 256 locations anywhere in the memory map.

Indirect Modes

Short Indirect. In this mode, the second byte of the instruction is used as a page zero address. The content of this page zero memory location is the LSB of the effective address. The MSBEA is zero (i.e. the effective address points to a page zero location).

Long Indirect. In the long indirect mode, the second byte of the instruction is used as a page zero pointer. The MSBEA is the content of this location while the LSBEA is the content of the following page zero location.

Indirect Indexed Modes

Short Indirect Indexed. The second byte of the instruction is used as a page zero address. To obtain the LSBEA, the content of this page zero memory location is added to the 8- bit unsigned value contained in the specified index register. The MSBEA is zero (i.e. the effective address points to a page zero location).

Long Indirect Indexed. In long indirect indexed mode, the second byte of the instruction is used as a page zero pointer. The 16-bit word read from this location (MSB at the pointed address and LSB at the following one) is added to the 8-bit unsigned value contained in the specified index register to form the 16-bit EA.

Relative Modes

Simple Relative. The relative addressing mode is used for branch instruction (e.g. Branch on bit, Branch on condition, Branch subroutine). The branch address (new value of PC) is calculated by adding a displacement given by the 8-bit signed byte following the op-code value to the actual content of the PC. This means that the variation of PC value is in the range -126 to +129 from the op-code address (the offset value can be calculated by the assembler).

Indirect Relative. The indirect relative addressing mode is similar to the relative mode but the content of the second byte of the instruction is used as a page zero address containing the 8-bit signed displacement value to be added to the actual content of the PC (i.e. address of the op-code plus 2).

SOFTWARE DESCRIPTION (Continued)

Direct Bit Modes

Bit Set And Clear. Bit Set/Clear mode is used to read-modify-write one single bit of a memory location in page zero, including I/O bits. The concerned memory location is given by the byte following the op-code (direct addressing mode) while the position of the bit to deal with is given in 3 bits included in the op-code.

Bit Test And Branch. Bit test and branch mode gives a relative branch according to the value of a single bit of a memory location in page zero. The op-code contains 3 bits to define the bit to test at a location given by the byte immediately following the op-code. A third byte gives the 8-bit signed value of the PC-relative displacement. If the test is true, this displacement is added to the actual content of the PC (i.e. op-code address plus 2) to form the new value of the PC.

Indirect Bit Modes

Indirect Bit Set And Clear. Indirect Bit Set/Clear mode works similarly to the bit Set/Clear mode except that the address of the concerned byte is the content of the location pointed by the second byte of the instruction.

Indirect Bit Test And Branch. This mode works as the bit test and branch mode but the tested byte is the content of the location pointed by the second byte of the instruction.

SOFTWARE DESCRIPTION (Continued)

Example

FUNCTION	SOURCE CODING	ADDRESSING MODES
		Immediate
Load a with memory	LD a,<ea >	$_2$A6 ^{2}

# Bytes	OP-Code	# Cycles

Table 10. Register/Memory and Absolute Jump Group (Group 1)

FUNCTION	SOURCE CODE	ADDRESSING MODE					
		Immediate	Direct	Extended	Index 0	Index 8	Index 16
Load A with memory	LD a,<ea>	$_2$A6 ^{2}	$_2$B6 ^{3}	$_3$C6 ^{4}	$_1$F6 ^{3}	$_2$E6 ^{4}	$_3$D6 ^{5}
Load iX with memory	LD iX,<ea>	$_2$AE ^{2}	$_2$BE ^{3}	$_3$CE ^{4}	$_1$FE ^{3}	$_2$EE ^{4}	$_3$DE ^{5}
Load memory with A	LD <ea>,a	—	$_2$B7 ^{4}	$_3$C7 ^{5}	$_1$F7 ^{4}	$_2$E7 ^{5}	$_3$D7 ^{6}
Load memory with iX	LD <ea>,iX	—	$_2$BF ^{4}	$_3$CF ^{5}	$_1$FF ^{4}	$_2$EF ^{5}	$_3$DF ^{6}
Add memory to A	ADD a,<ea>	$_2$AB ^{2}	$_2$BB ^{3}	$_3$CB ^{4}	$_1$FB ^{3}	$_2$EB ^{4}	$_3$DB ^{5}
Add memory and carry to A	ADC a,<ea>	$_2$A9 ^{2}	$_2$B9 ^{3}	$_3$C9 ^{4}	$_1$F9 ^{3}	$_2$E9 ^{4}	$_3$D9 ^{5}
Subtract memory to A	SUB a,<ea>	$_2$A0 ^{2}	$_2$B0 ^{3}	$_3$C0 ^{4}	$_1$F0 ^{3}	$_2$E0 ^{4}	$_3$D0 ^{5}
Subtract memory with carry	SBC a,<ea>	$_2$A2 ^{2}	$_2$B2 ^{3}	$_3$C2 ^{4}	$_1$F2 ^{3}	$_2$E2 ^{4}	$_3$D2 ^{5}
And memory to A	AND a,<ea>	$_2$A4 ^{2}	$_2$B4 ^{3}	$_3$C4 ^{4}	$_1$F4 ^{3}	$_2$E4 ^{4}	$_3$D4 ^{5}
Or memory with A	OR a,<ea>	$_2$AA ^{2}	$_2$BA ^{3}	$_3$CA ^{4}	$_1$FA ^{3}	$_2$EA ^{4}	$_3$DA ^{5}
Exclusive OR	XOR a,<ea>	$_2$A8 ^{2}	$_2$B8 ^{3}	$_3$C8 ^{4}	$_1$F8 ^{3}	$_2$E8 ^{4}	$_3$D8 ^{5}
Arithmetic Compare A	CP a,<ea>	$_2$A1 ^{2}	$_2$B1 ^{3}	$_3$C1 ^{4}	$_1$F1 ^{3}	$_2$E1 ^{4}	$_3$D1 ^{5}
Arithmetic Compare iX	CP iX,<ea>	$_2$A3 ^{2}	$_2$B3 ^{3}	$_3$C3 ^{4}	$_1$F3 ^{3}	$_2$E3 ^{4}	$_3$D3 ^{5}
Bit compare A and memory	BCP a,<ea>	$_2$A5 ^{2}	$_2$B5 ^{3}	$_3$C5 ^{4}	$_1$F5 ^{3}	$_2$E5 ^{4}	$_3$D5 ^{5}
Absolute Jump	JP <ea>	—	$_2$BC ^{2}	$_3$CC ^{3}	$_1$FC ^{2}	$_2$EC ^{3}	$_3$DC ^{4}
Call subroutine	CALL <ea>	—	$_2$BD ^{5}	$_3$CD ^{6}	$_1$FD ^{5}	$_2$ED ^{6}	$_3$DD ^{7}

SOFTWARE DESCRIPTION (Continued)

Table 11. Read - Modify - Write Group (Group 2)

FUNCTION	SOURCE CODE	ADDRESSING MODE						
		Inh a	Inh ix	Direct	Memory Direct	Index 0	Index +d8	Index +[ad8]
Increment	INC <ea>	1 4C [3]	1 5C [3]	2 3C [5]	3 923C [7]	1 7C [5]	2 6C [6]	3 926C [8]
(Y index)			2 905C [4]			2 907C [6]	3 906C [7]	3 916C [8]
Decrement	DEC <ea>	1 4A [3]	1 5A [3]	2 3A [5]	3 923A [7]	1 7A [5]	2 6A [6]	3 926A [8]
(Y index)			2 905A [4]			2 907A [6]	3 906A [7]	3 916A [8]
Clear	CLR <ea>	1 4F [3]	1 5F [3]	2 3F [5]	3 923F [7]	1 7F [5]	2 6F [6]	3 926F [8]
(Y index)			2 905F [4]			2 907F [6]	3 906F [7]	3 916F [8]
One's Complement	CPL <ea>	1 43 [3]	1 53 [3]	2 33 [5]	3 9233 [7]	1 73 [5]	2 63 [6]	3 9263 [8]
(Y index)			2 9053 [4]			2 9073 [6]	3 9063 [7]	3 9163 [8]
Negate (2's complement)	NEG <ea>	1 40 [3]	1 50 [3]	2 30 [5]	3 9230 [7]	1 70 [5]	2 60 [6]	3 9260 [8]
(Y index)			2 9050 [4]			2 9070 [6]	3 9060 [7]	3 9160 [8]
Rotate Left thru Carry	RLC <ea>	1 49 [3]	1 59 [3]	2 39 [5]	3 9239 [7]	1 79 [5]	2 69 [6]	3 9269 [8]
(Y index)			2 9059 [4]			2 9079 [6]	3 9069 [7]	3 9169 [8]
Rotate Right thru Carry	RRC <ea>	1 46 [3]	1 56 [3]	2 36 [5]	3 9236 [7]	1 76 [5]	2 66 [6]	3 9266 [8]
(Y index)			2 9056 [4]			2 9076 [6]	3 9066 [7]	3 9166 [8]
Shift Left Logical	SLL <ea>	1 48 [3]	1 58 [3]	2 38 [5]	3 9238 [7]	1 78 [5]	2 68 [6]	3 9268 [8]
(Y index)			2 9058 [4]			2 9078 [6]	3 9068 [7]	3 9168 [8]
Shift Right Logical	SRL <ea>	1 44 [3]	1 54 [3]	2 34 [5]	3 9234 [7]	1 74 [5]	2 64 [6]	3 9264 [8]
(Y index)			2 9054 [4]			2 9074 [6]	3 9064 [7]	3 9164 [8]
Shift Left Arithmetic	SLA <ea>	1 48 [3]	1 58 [3]	2 38 [5]	3 9238 [7]	1 78 [5]	2 68 [6]	3 9268 [8]
(Y index)			2 9058 [4]			2 9078 [6]	3 9068 [7]	3 9168 [8]
Shift Right Arithmetic	SRA <ea>	1 47 [3]	1 57 [3]	2 37 [5]	3 9237 [7]	1 77 [5]	2 67 [6]	3 9267 [8]
(Y index)			2 9057 [4]			2 9077 [6]	3 9067 [7]	3 9167 [8]
Test for Negative or Zero	TNZ <ea>	1 4D [3]	1 5D [3]	2 3D [4]	3 923D [6]	1 7D [4]	2 6D [5]	3 926D [7]
(Y index)			2 905D [4]			2 907D [5]	3 906D [6]	3 916D [7]
Swap Nibbles	SWAP <ea>	1 4E [3]	1 5E [3]	2 3E [5]	3 923E [7]	1 7E [5]	2 6E [6]	3 926E [8]
(Y index)			2 905E [4]			2 907E [6]	3 906E [7]	3 916E [8]

SOFTWARE DESCRIPTION (Continued)

Table 12. Bit Manipulation And Test Group (Group 3)

FUNCTION	SOURCE CODE	ADDRESSING MODES	
		Relative	Indirect Relative
Bit Set	BSET < ea > , # b	2 (10+2*b) [5]	3 92(10+2*b) [7]
Bit Reset	BRES < ea > , # b	2 (11+2*b) [5]	3 92(10+2*b) [7]
Bit Test and Jump if True	BTJT < ea > , # b , ee	3 (00+2*b) [5]	4 92(00+2*b) [7]
Bit Test and Jump if False	BTJF < ea > , # b , ee	3 (01+2*b) [5]	4 92(01+2*b) [7]

Table 13. PC-Relative Jump Group (Group 4)

FUNCTION	SOURCE CODE		ADDRESSING MODES	
			Relative	Indirect Relative
Jump Relative True	JRT	ee	2 20 [3]	3 9220 [5]
(Jump Relative always)	JRA	ee	2 20 [3]	3 9220 [5]
Jump Relative False	JRF	ee	2 21 [3]	3 9221 [5]
Jump Relative if Unsigned Greater than	JRUGT	ee	2 22 [3]	3 9222 [5]
Jump Relative if Unsigned Lower or Equal	JRULE	ee	2 23 [3]	3 9223 [5]
Jump Relative if No Carry	JRNC	ee	2 24 [3]	3 9224 [5]
Jump Relative if Unsigned Greater or Equal	JRUGE	ee	2 24 [3]	3 9224 [5]
Jump Relative if Carry	JRC	ee	2 25 [3]	3 9225 [5]
Jump Relative if Unsigned Lower than	JRULT	ee	2 25 [3]	3 9225 [5]
Jump Relative if Not Equal	JRNE	ee	2 26 [3]	3 9226 [5]
Jump Relative if Equal	JREQ	ee	2 27 [3]	3 9227 [5]
Jump Relative if Half Carry	JRH	ee	2 28 [3]	3 9228 [5]
Jump Relative if Not Half Carry	JRNH	ee	2 29 [3]	3 9229 [5]
Jump Relative if Plus	JRPL	ee	2 2A [3]	3 922A [5]
Jump Relative if Minus	JRMI	ee	2 2B [3]	3 922B [5]
Jump Relative if Not Interrupt Mask	JRNM	ee	2 2C [3]	3 922C [5]
Jump Relative if Interrupt Mask	JRM	ee	2 2D [3]	3 922D [5]
Jump Relative if Interrupt Line Low	JRIL	ee	2 2E [3]	3 922E [5]
Jump Relative if Interrupt Line High	JRIH	ee	2 2F [3]	3 922F [5]
Call Subroutine Relative	CALLR	ee	2 AD [6]	3 92AD [8]

SOFTWARE DESCRIPTION (Continued)

Table 14. Miscellaneous Group(Group 5)

FUNCTION	SOURCE CODE	X iNDEX
Multiply (iX, A = iX * A)	MUL iX , a	1 42 [11]
Load iX with acc. a content	LD iX , a	1 97 [2]
Load a with iX content	LD a , iX	1 9F [2]
Load Stack p. with acc. a content	LD S , a	1 95 [2]
Load acc. a with Stack p. content	LD a , S	1 9E [2]
Load Stack p. with iX content	LD S , iX	1 94 [2]
Load iX with Stack p. content	LD iX , S	1 96 [2]
Load X // with Y // content	LD X , Y	1 93 [2]
Load Y // with X // content	LD Y , X	—
Push acc. a onto the Stack	PUSH A	1 88 [3]
Pop acc. a from the Stack	POP A	1 84 [4]
Push iX onto the stack	PUSH iX	1 89 [3]
Pop iX from the Stack	POP iX	1 85 [4]
Push Condition Codes onto the Stack	PUSH CC	1 8A [3]
Pop Condition Codes from the Stack	POP CC	1 86 [4]
Reset Carry Flag	RCF	1 98 [2]
Set Carry Flag	SCF	1 99 [2]
Reset Interrupt Mask	RIM	1 9A [2]
Set Interrupt Mask	SIM	1 9B [2]
Reset Stack Pointer	RSP	1 9C [2]
No Operation	NOP	1 9D [2]
Interrupt Routine Return	IRET	1 80 [9]
Subroutine Return	RET	1 81 [6]
Software Trap	TRAP	1 83 [10]
Halt	HALT	1 8E [2]
Wait For Interrupt	WFI	1 8F [2]

SOFTWARE DESCRIPTION (Continued)

ST7271 ASSEMBLER REGISTER DEFINITION INCLUDE FILE

```
            BYTES       segment byte at 00-4F 'periph'

.PADR       ds.b        1       ; Port A data register
.PBDR       ds.b        1       ; Port B data register
.PCDR       ds.b        1       ; Port C data register
.PDDR       ds.b        1       ; Port D data register
.PADDR      ds.b        1       ; Port A data direction reg
.PBDDR      ds.b        1       ; Port B data direction reg
.PCDDR      ds.b        1       ; Port C data direction reg
.PDDDR      ds.b        1       ; Port D data direction reg
.ADCDR      ds.b        1       ; ADC data register (port B)
.ADCCR      ds.b        1       ; ADC control register
            ds.b        3       ; reserved
.EWPCC0     ds.b        1       ; East/West control 0
.EWPCC1     ds.b        1       ; East/West control 1
.EEP0CR     ds.b        1       ; E/W EEPROM control reg
.EEP1CR     ds.b        1       ; EEPROM 1 control reg
.EEP2CR     ds.b        1       ; EEPROM 2 control reg
.TIMCR      ds.b        1       ; Timer control register
.TIMSR      ds.b        1       ; Timer status register
.IC1HR      ds.b        1       ; Input capture 1 high reg
.IC1LR      ds.b        1       ; Input capture 2 low reg
.OC1HR      ds.b        1       ; Output compare1 high reg
.OC1LR      ds.b        1       ; Output compare1 low reg
.CNTHR      ds.b        1       ; Counter high register
.CNTLR      ds.b        1       ; Counter low register
.ACNTHR     ds.b        1       ; Alternate cnt high reg
.ACNTLR     ds.b        1       ; Alternate cnt low reg
.IC2HR      ds.b        1       ; Input capture 2 high reg
.IC2LR      ds.b        1       ; Input capture 2 low reg
.OC2HR      ds.b        1       ; Output compare 2 high reg
.OC2LR      ds.b        1       ; Output compare 2 low reg
.PWM0       ds.b        1       ; PWM0 register
.BRM0       ds.b        1       ; BRM0 register (12 bits)
.PWM1       ds.b        1       ; PWM1 register
.BRM1       ds.b        1       ; BRM1 register (12 bits)
.PWM2       ds.b        1       ; PWM2 register
.BRM2_3     ds.b        1       ; BRM2&3 register
.PWM3       ds.b        1       ; BRM3 register
```

SOFTWARE DESCRIPTION (Continued)

```
.PWM4        ds.b        1      ; PWM4 register
.BRM4_5      ds.b        1      ; BRM4&5 register
.PWM5        ds.b        1      ; BRM5 register
.PWM6        ds.b        1      ; PWM6 register
.BRM6_7      ds.b        1      ; BRM6&7 register
.PWM7        ds.b        1      ; BRM7 register
.PWM8        ds.b        1      ; PWM8 register
.BRM8_9      ds.b        1      ; BRM8&9 register
.PWM9        ds.b        1      ; BRM9 register
.PWM10       ds.b        1      ; PWM10 register
.BRM10_11    ds.b        1      ; BRM10&11 register
.PWM11       ds.b        1      ; BRM11 register
.PWM12       ds.b        1      ; PWM12 register
.BRM12_13    ds.b        1      ; BRM12&13 register
.PWM13       ds.b        1      ; BRM13 register
.PWM14       ds.b        1      ; PWM14 register
.BRM14_15    ds.b        1      ; BRM14&15 register
.PWM15       ds.b        1      ; BRM15 register
.PWM16       ds.b        1      ; PWM16 register
.BRM16_17    ds.b        1      ; BRM16&17 register
.PWM17       ds.b        1      ; BRM17 register
.SYNCMCR     ds.b        1      ; Sync processor MCR
.SYNCCCR     ds.b        1      ; Sync processor CCR
.EWDAC       ds.b        1      ; Ewpcc DAC register
             ds.b        1      ; reserved
.SPIDTAIOR   ds.b        1      ; SPI register not used
             ds.b        1      ; reserved
.SPICTRLR    ds.b        1      ; SPI control register
.SPISTSR     ds.b        1      ; SPI status register
.MISCR       ds.b        1      ; Miscelleanous reg
.PBICFGR     ds.b        1      ; Port B input config reg
.PRGIOCFGR   ds.b        1      ; Programmable I/O config

             segment byte at 50-BF 'ram'
             segment byte at C0-FF 'stack'
             WORDS
             segment byte at 100-14F 'ram2'
             segment byte at 150-44F 'eeprom'
             segment byte at 450-3EFF 'rom'
             segment byte at 3FF0-3FFF 'vectit'
```

3.2 ELECTRICAL CHARACTERISTICS

The ST7271 device contains circuitry to protect the inputs againts damage due to high static voltage or electric field. Nevertheless it is advised to take normal precautions and to avoid applying to this high impedance voltage circuit any voltage higher than the maximum rated voltages. It is recommended for proper operation that V_{IN} and V_{OUT} be constrained to the range :

$V_{SS} \leq V_{IN}$ or $V_{OUT} \leq V_{DD}$

To enhance reliability of operation, it is recommended to connect unused inputs to an appropriate logic voltage level such as V_{SS} or V_{DD}.

All the voltage in the following tables are referenced to V_{SS}

ABSOLUTE MAXIMUM RATINGS (Voltage Referenced to Vss)

Symbol	Ratings	Value	Unit
V_{DD}	Supply Voltage	-0.3 to +7.0	V
V_{IN}	Input Voltage	V_{SS} -0.3 to V_{DD} + 0.3	V
V_{DDA}	Analog Reference Voltage	-0.3 to +9.0	V
I	Current Drain Per Pin Excluding V_{DD} and V_{SS}	25	mA
T_A	Operating Temperature Range Suffix Option 1 (Standard)	T_L to T_H 0 to +70	°C
T_{STG}	Storage Temperature Range	-65 to +150	°C

ELECTRICAL CHARACTERISTICS (Continued)

T_J, the average chip-junction temperature in Celsius can be calculated from the following equation:

$$T_J = T_A + (P_D \times \theta JA) \quad (1)$$

Where:

- T_A is the Ambient Temperature in °C,
- qJA ithe Package Thermal Resistance, Junction-to-Ambient in °C/W,
- P_D the sum of P_{INT} and $P_{I/O}$,
- P_{INT} equals I_{CC} time V_{CC}, Watts-Chip Internal Power
- $P_{I/O}$ the Power Dissipation on Input and Output Pins, User Determined.

For most applications $P_{I/O} < P_{INT}$ and can be neglected .

P_{PORT} may be significant if the device is configured to drive Darlington bases or sink LED Loads.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Therefore :

$$K = P_D \times (T_A + 273°C) + \theta J_A \times P_D^2 \quad (3)$$

Where K is constant pertaining to the particular part, K can be determined from equation (3) by measuring P_D (at equilibrium) for a kwon T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A

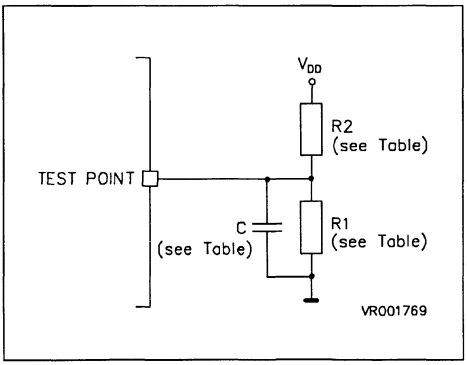
Thermal Characteristics

Symbol	Characteristics	Value	Unit
θJA	Thermal Resistance PSDIP56 PSDIP42		°C/W

Equivalent Test Load

V_{DD} = 4.5V			
Pins	**R1**	**R2**	**C**
PA0-PA7	3.26kΩ	2.38kΩ	50pF
PB0-PB7	3.26kΩ	2.38kΩ	50pF
PC0-PC5	3.26kΩ	2.38kΩ	50pF
PD0-PD4	3.26kΩ	2.38kΩ	50pF

Test Diagram



ELECTRICAL CHARACTERISTICS (Continued)

AC Electrical Characteristics

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{DD}	Operating Supply Voltage	RUN Mode STOP Mode EEPROM Write EEPROM Read	4.5 4.5 4.5 4.5		5.5	V
V_{DDA}	Analog Reference Voltage		7.4	8	8.6	V
I_{DD}	Supply Current [1]					
		RUN Mode V_{DD}=5V,f_{OSC}=8MHz		1.8	2.5	mA
		WAIT Mode V_{DD}=5V,f_{OSC}=8MHz		0.9	1.5	mA
		EEPROM Programming V_{DD}=5V,f_{OSC}=8MHz			2.5	µA
		STOP Mode V_{DD}=5V,T_A=70°C		1	10	µA
V_{OL}	Output Low Voltage	Port A, B - Open Drain I_{LOAD} =200µA			0.1xV_{DD}	V

Note 1. Measured with a quartz crystal for the 8 MHz Frequency.

ELECTRICAL CHARACTERISTICS (Continued)

DC Electrical Characteristics

(V_{DD} = 5.0V±10%, V_{SS} = 0V, T_A = operating temperature range, unless otherwise noted)

Symbol	Test Conditions	Min.	Typ.	Max.	Unit
V_{OL}	Output Voltage, Load ≤ 10.0µA			0.1	V
V_{OH}		V_{DD}-0.1			V
V_{OH}	Output High Voltage				
	I_{LOAD} = 0.8mA, PA0-PA7,PB0-PB7,PC0-PC5, PD0-PD4	V_{DD}-0.8			V
V_{OL}	Output Low Voltage				V
	(I_{LOAD} = 1.6 mA) PA0-PA7,PB0-PB7,PC0-PC5, PD0-PD4, $\overline{RESET}$			0.4	
V_{IH}	Input High Voltage				
	PA0-PA7,PB0-PB7,PC0-PC5, PD0-PD4, $\overline{RESET}$	0.7xV_{DD}		V_{DD}	V
V_{IL}	Input Low Voltage				
	PA0-PA7,PB0-PB7,PC0-PC5, PD0-PD4, $\overline{RESET}$	V_{SS}		0.2xV_{DD}	V
V_{RM}	Data Retention Mode (0 to 70°C)	2			V
I_{IL}	I/O Ports Hi-Z Leakage Current				µA
	PA0-PA7,PB0-PB7,PC0-PC5, PD0-PD4			± 10	
I_{IN}	Input Current : $\overline{RESET}$			± 1	µA
C_{OUT}	Capacitance : Ports (as Input or Output)			12	pF
C_{IN}	$\overline{RESET}$			8	pF

ELECTRICAL CHARACTERISTICS (Continued)

Control Timing

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{OSC}	Frequency of Operation	V_{DD}=5V	DC		8	MHz
t_{ILCH}	STOP Mode Recovery Startup Time	Ceramic resonator			20	ms
t_{RL}	External RESET Input Pulse Width		1.5			t_{CYC}
t_{PORL}	Power RESET Ouput		4096			t_{CYC}
t_{DOGL}	Watchdog RESET Output Pulse Width		1.5			t_{CYC}
t_{DOG}	Watchdog Time-out		6144		7168	t_{CYC}
t_{PROG}	EEPROM Programming Time (0 to 70)			2	8	ms
t_{ILIH}	Interrupt Pulse Width PORTB		125			ns
t_{ILIL}	Interrupt Pulse Period		(1)			t_{CYC}
t_{DDR}	Power up rise time	V_{DDmin}			100	ms

Note 1. The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 cycles.

ELECTRICAL CHARACTERISTICS (Continued)

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

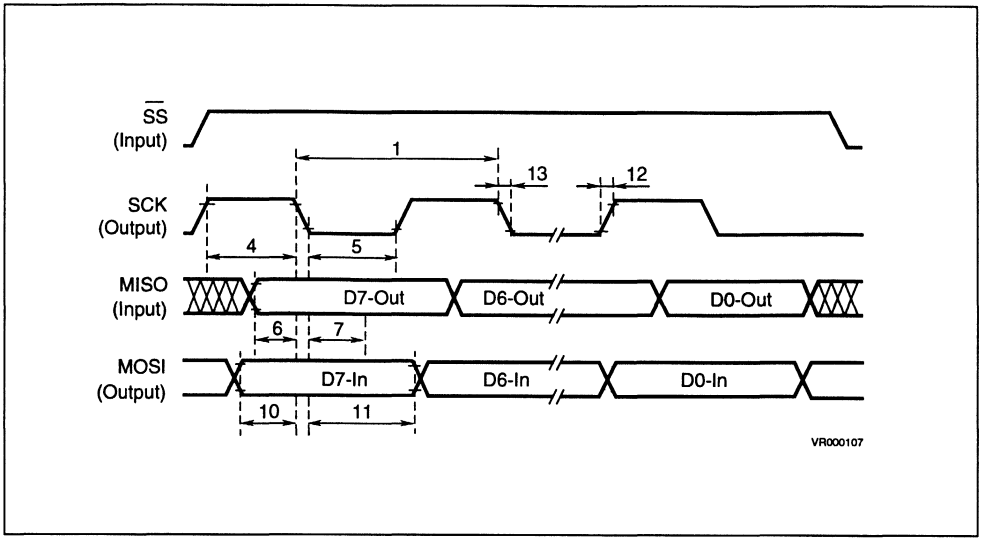
(V_{DD} = 5.0 V_{DC} ± 10%, V_{SS} = 0 V_{DC}, T_A = T_L to T_H)

Num.	Symbol	Characteristics	Fosc = 8.0MHz		Unit
			Min.	Max.	
		Operating Frequency = $F_{OSC/2}$ = F_{OP}			
	$f_{OP}(m)$	Master	dc	0.5	f_{OP}
	$f_{OP}(s)$	Slave	dc	4.0	MHz
1		Cycle Time			
	$t_{CYC(m)}$	Master	2.0		tCYC
	$t_{CYC(s)}$	Slave	240		ns
2		Enable Lead Time			ns
	$t_{lead(m)}$	Master	(1)		
	$t_{lead(s)}$	Slave	120		
3		Enable Lag time			ns
		Master	(1)		
		Slave	120		
4		Clock (SCK)High Time			
	$t_{W(SCKH)}$	Master	100		ns
	$t_{W(SCKH)}$	Slave	90		ns
5		Clock (SCK) Low Time			
	$t_{W(SCKL)}$	Master	100		ns
	$t_{W(SCKL)}$	Slave	90		ns
6		Data Set-up Time			
	$t_{SU(m)}$	Master	100		ns
	$t_{SU(s)}$	Slave	100		ns
7		Data Hold Time (Inputs)			
	$t_{H(m)}$	Master	100		ns
	$t_{H(s)}$	Slave	100		ns
8	t_A	Access Time (Time to Data Active from High Impedance State)			ns
		Slave	0	120	
9	t_{DIS}	Disable Time (Hold Time to High Impedance State)			ns
		Slave		240	
10		Data Valid			
	$t_{V(m)}$	Master (Before Capture Edge)	0.25		$t_{CYC(m)}$
	$t_{V(s)}$	Slave (After Enable Edge) (2)		120	ns
11		Data Hold Time (Outputs)			
	$t_{HO(m)}$	Master (Before Capture Edge)	0.25		$t_{CYC(m)}$
	$t_{HO(s)}$	Slave (After Enable Edge)	0		ns
12		Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200pF)			
	t_{RM}	SPI Outputs (SCK, MOSI, MISO)		100	ns
	t_{RS}	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)		2.0	µs
13		Fall Time (70% V_{DD} to 20% V_{DD}, C_L			
	t_{FM}	SPI Outputs (SCK, MOSI, MISO)		100	ns
	t_{FS}	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)		2.0	µs

Notes: 1 Signal production depends on software. 2 Assumes 200pF load on all SPI pins.

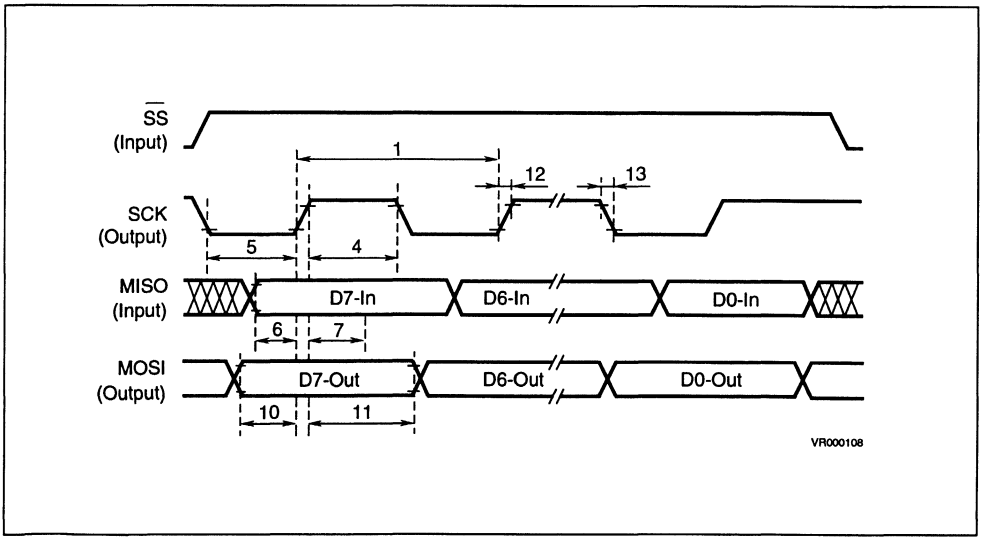
ELECTRICAL CHARACTERISTICS (Continued)

Figure 43. SPI Master Timing Diagram CPOL=0, CPHA=1



Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

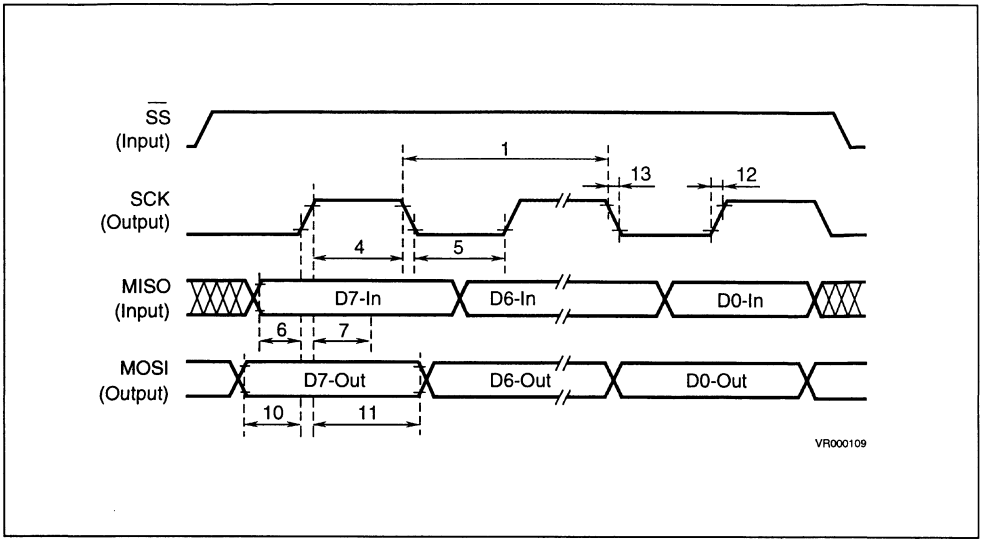
Figure 44. SPI Master Timing Diagram CPOL=1, CPHA=1



Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

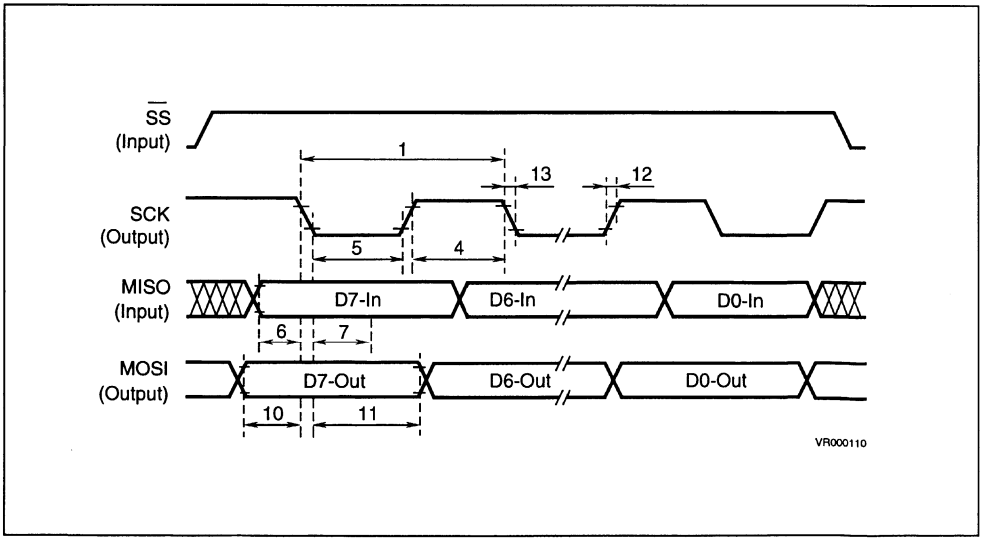
ELECTRICAL CHARACTERISTICS (Continued)

Figure 45. SPI Master Timing Diagram CPOL=0, CPHA=0



Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

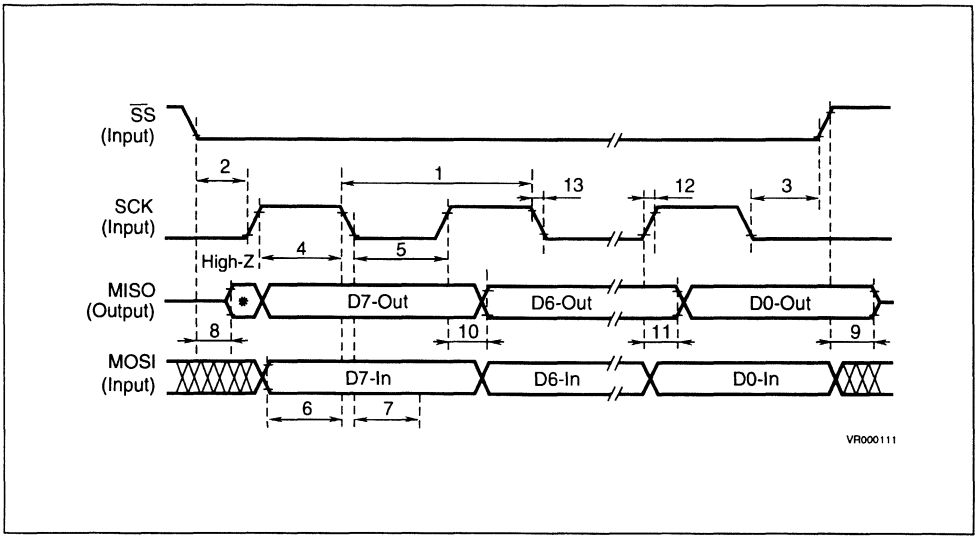
Figure 46. SPI Master Timing Diagram CPOL=1, CPHA=0



Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

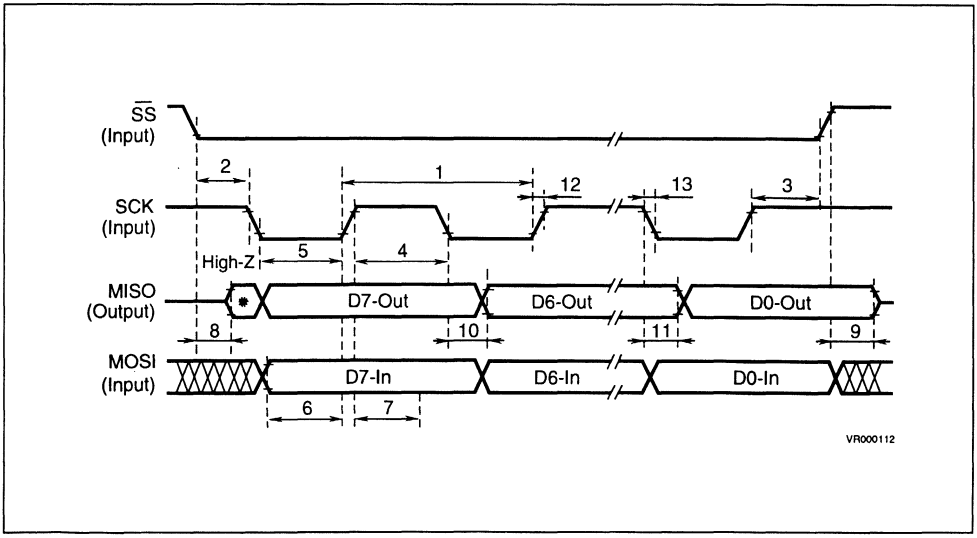
ELECTRICAL CHARACTERISTICS (Continued)

Figure 47. SPI Slave Timing Diagram CPOL=0, CPHA=1



Notes: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}
* Denotes undefined, either high or low .

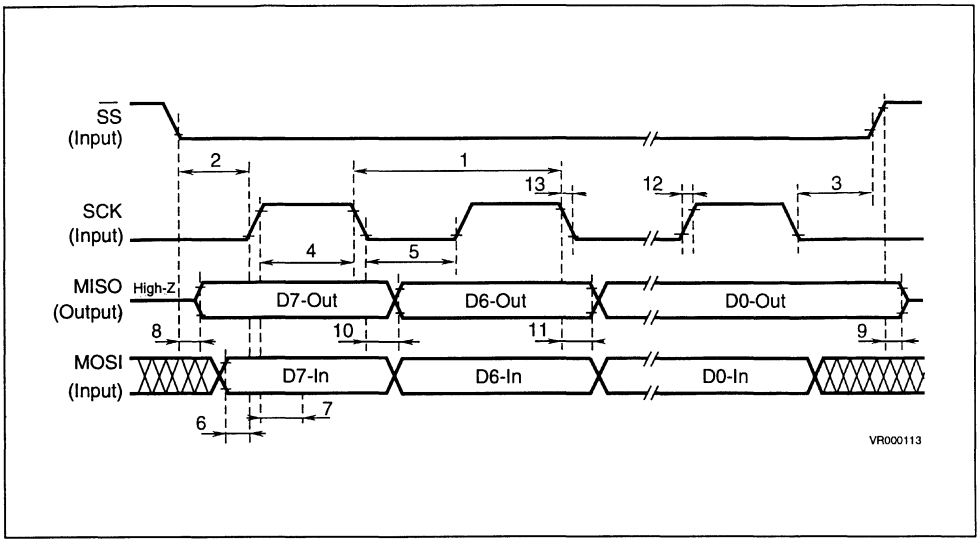
Figure 48. SPI Slave Timing Diagram CPOL=1, CPHA=1



Notes: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}
* Denotes undefined, either high or low .

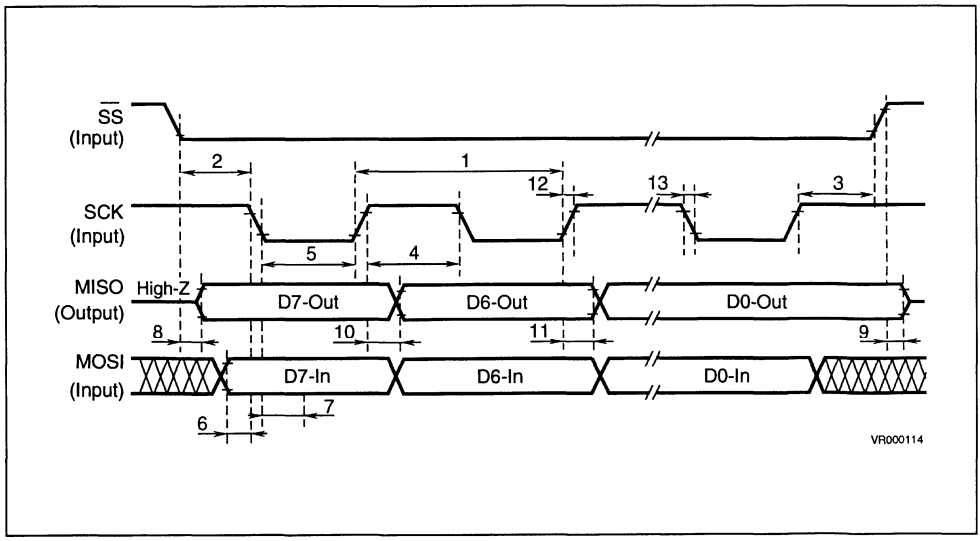
ELECTRICAL CHARACTERISTICS (Continued)

Figure 49. SPI Slave Timing Diagram CPOL=0, CPHA=0



Notes: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}
* Denotes undefined, either high or low .

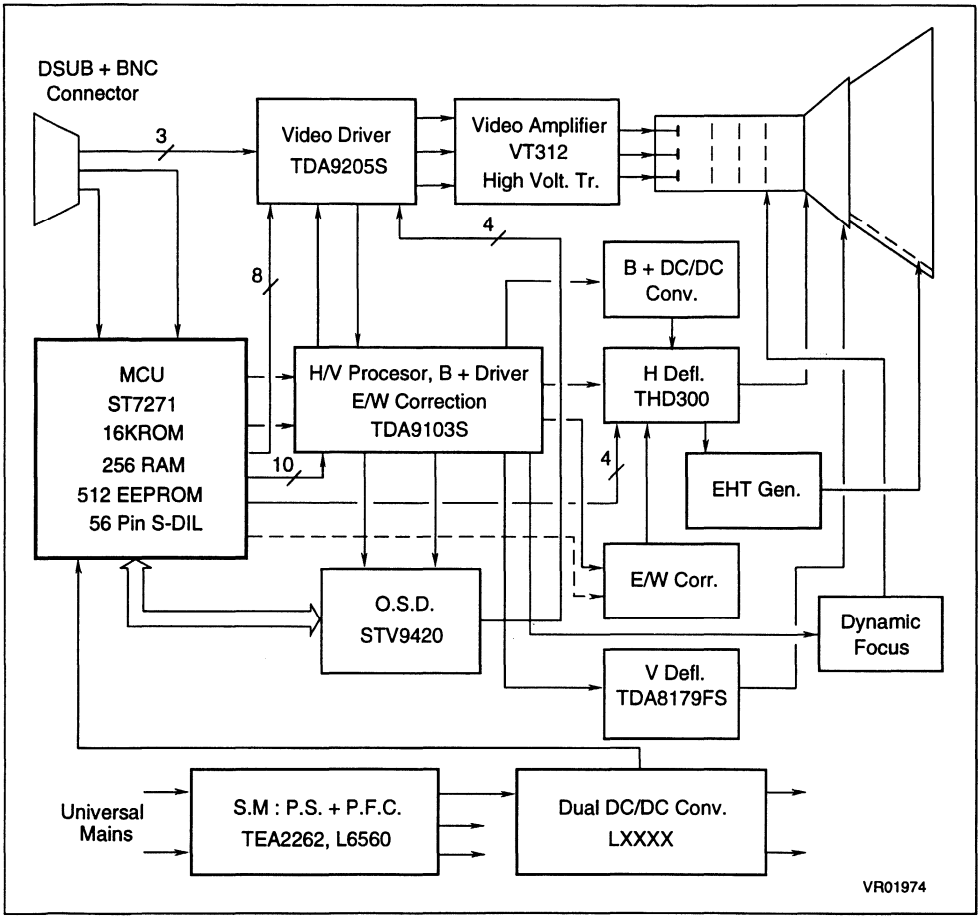
Figure 50. SPI Slave Timing Diagram CPOL=1, CPHA=0



Notes: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}
* Denotes undefined, either high or low .

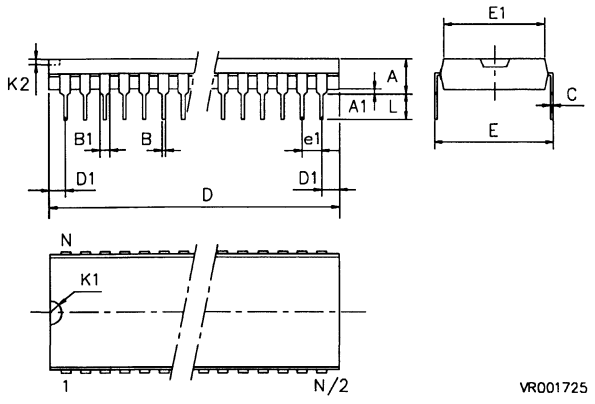
3.3 APPLICATION EXAMPLE

Figure 51. MultiSync 17" to 21" Monitor Block Diagram (with OSD and Dynamic Focus Control)



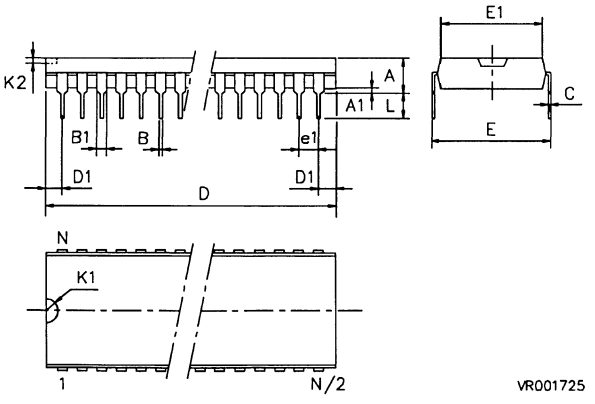
3.4 PACKAGE MECHANICAL DATA

Figure 52. 56-Shrink Plastic Dual In line Package, 600-Mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D		52.12			2.052	
D1	–	–	–	–	–	–
E	15.2 BSD			0.600 BSD		
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	56					

Figure 53. 42-Shrink Plastic Dual In line Package, 600-Mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D	36.32		39.12	1.430		1.540
D1	–	–	–	–	–	–
E	15.2 BSD			0.600 BSD		
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	42					

3.5 ORDERING INFORMATION

Ordering Information Table

Sales Type	ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	Temperature Range	Package
ST7271N5B1	16K	256	512	-0 to + 70°C	PSDIP56
ST7271N3B1	12K	256	512		PSDIP56
ST7271N1B1	8K	192	384		PSDIP56
ST7271J1B1	8K	192	384		PSDIP42

ST7271 MICROCONTROLLER OPTION LIST

Customer: .....
Address: .....
Contact: .....
Phone No: .....
Reference: .....

SGS-THOMSON Microelectronics references

Device:

PSDIP56	[ ] ST7271N5	[ ] ST7271N3	[ ] ST7271N1
PSDIP42	[ ] ST7271J1		

Temperature Range	[ ] 0 to 70°C	
Software Development	[ ] SGS-THOMSON	[ ] Customer
	[ ] External Laboratory	

For marking one line with 11 characters maximum is possible

Special Marking [ ] No
[ ] Yes "_ _ _ _ _ _ _ _ _ _ _"
Letters, digits, ' .', ' - ', '/ ' and spaces only

OPTION LIST:

Watchdog State After Reset	[ ] Enable	[ ] Disable
Watchdog during WAITmode	[ ] Active	[ ] Suspend
Input Clock to SPI (8MHz osc)	[ ] 2MHz	[ ] 4MHz

Signature .....

Date .....

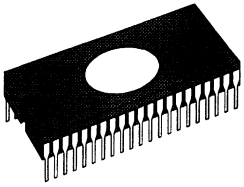
SGS-THOMSON MICROELECTRONICS

ST72E71
ST72T71

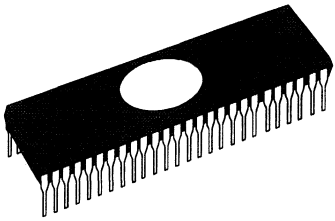
8-BIT EPROM HCMOS MCUs WITH EEPROM AND TV/MONITOR DEDICATED FUNCTIONS

PRELIMINARY DATA

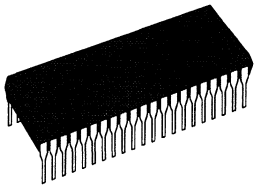
- 5V ± 10% supply operating range
- 4MHz Maximum Internal Clock Frequency
- Fully static operation
- 0 to +70°C Operating Temperature Range
- Run, Wait, and Stop Modes
- User EPROM: up to 15144 bytes
  Data RAM: up to 256 bytes
  EEPROM: up to 512 bytes
  EWPCC EEPROM: 256 bytes
- 56 pin Windowed Ceramic Shrink Dual In line Package (ST7271N)
  42 pin Windowed Ceramic Shrink Dual In Line Package (ST7271J)
- up to 27 I/O lines
  8 I/O Open Drain with 12V capability
- up to 8 lines programmable as interrupt wake-up inputs
- 16-bit timer with 2 input capture and 2 output compare functions
- Sync Processor for video timing analysis
- East/West Pin Cushion Automatic Correction with DAC output.
- Watchdog for system reliability and integrity
- 8-bit Analog to Digital Converter with up to 8 channels
- 16 10-bit PWM/BRM Digital to Analog outputs
- 2 12-bit PWM/BRM Digital to Analog outputs
- Industry Standard Serial Peripheral Interface
- User mask options:
  - SPI Data Rate
  - Watchdog enable/disable after Reset
  - Watchdog enable during WAIT mode
- Master Reset and Power-on reset
- Full Hardware Emulator
- 8-bit data manipulation
- 74 basic instructions
- 10 main addressing modes
- 8x8 unsigned multiply instruction
- true bit manipulation
- Complete development support on Real-time emulator with PC/DOS
- Full software package (Cross Assembler, debugger)



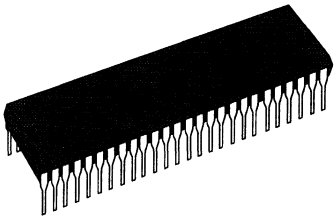
CSDIP42W



CSDIP56W



PSDIP42



PSDIP56

(Ordering Information at the end of the datasheet)

December 1993

85/95

Figure 1a. 56 Pin Shrink DIP Pinout

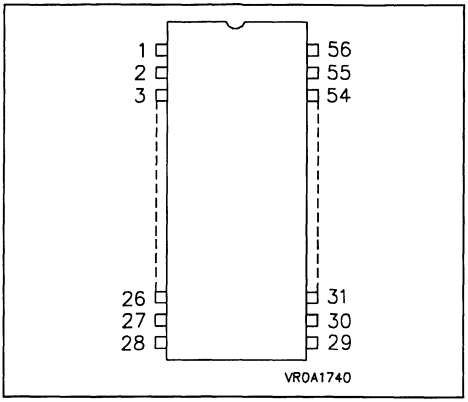
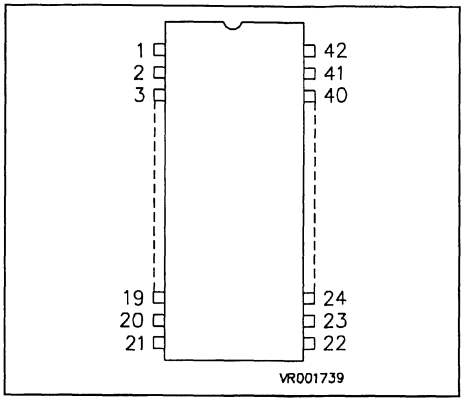


Figure 1b. 42 Pin Shrink DIP Pinout



Pin Description

Pin	Name	Pin	Name
1	V_{DDA}	56	V_{SSA}
2	EWPCC	55	V_{SS}
3	DA0	54	PC5/$\overline{SS}$
4	DA1	53	PC4/SCK
5	DA2	52	PC3/MOSI
6	DA3	51	PC2/MISO
7	DA4	50	PC1
8	DA5	49	PC0/OCMP
9	DA6	48	TEST/V_{PP}
10	DA7	47	DA17
11	DA8	46	DA16
12	DA9	45	DA15
13	PB7	44	DA14
14	PB6	43	PA0
15	PB5	42	PA1
16	PB4	41	PA2
17	PB3	40	PA3
18	PB2	39	PA4
19	PB1	38	PA5
20	VFBACK/PB0	37	PA6
21	PD4	36	PA7
22	CLMPO/PD3	35	DA13
23	DA10	34	DA12
24	DA11	33	OSCIN
25	$\overline{RESET}$	32	OSCOUT
26	VSYNCO/PD2	31	CSYNCI/PDO
27	VSYNCI	30	HSYNCO/PD1
28	V_{DD}	29	HSYNCI

Pin Description

Pin	Name	Pin	Name
1	V_{DDA}	42	V_{SSA}
2	EWPCC	41	V_{SS}
3	DA0	40	PC5/$\overline{SS}$
4	DA1	39	PC4/SCK
5	DA2	38	PC3/MOSI
6	DA3	37	PC2/MISO
7	DA4	36	PC0/OCMP
8	DA5	35	TEST/V_{PP}
9	DA6	34	PA0
10	DA7	33	PA1
11	DA8	32	PA2
12	DA9	31	PA3
13	PB3	30	PA4
14	PB2	29	PA5
15	PB1	28	PA6
16	VFBACK/PB0	27	PA7
17	CLMPO/PD3	26	OSCIN
18	$\overline{RESET}$	25	OSCOUT
19	VSYNCO/PD2	24	CSYNCI/PDO
20	VSYNCI	23	HSYNCO/PD1
21	V_{DD}	22	HSYNCI

1 GENERAL DESCRIPTION

4.1 INTRODUCTION

The ST72E71,T71 is a HCMOS microcontroller unit (MCU) from the ST72 family with dedicated peripherals for TV and Monitor applications.

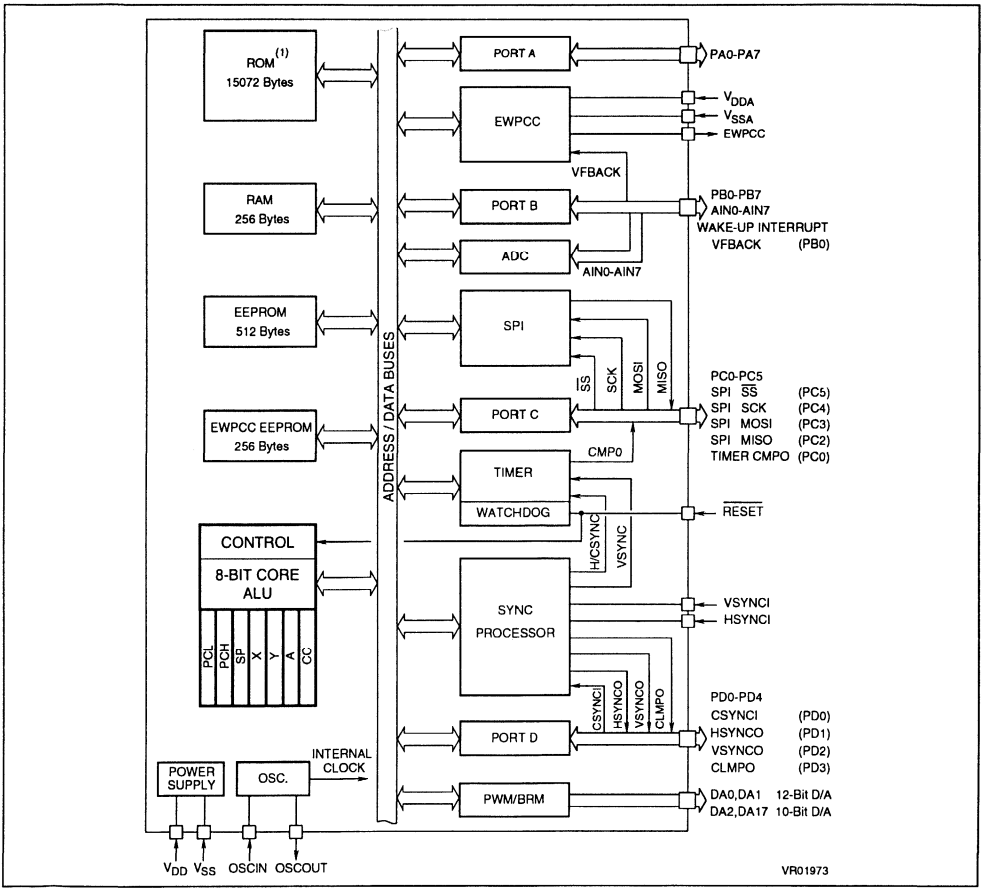
The ST72E71 is the EPROM version of the ST7271 ROM device, suitable for development. The ST72T71 is the OTP version, suitable for product prototyping and low volume production.

It is based around an industry standard 8-bit core and offers an enhanced instruction set. The processor runs with an external clock at 8 MHz with a 5V supply. Due to the fully static design of this device, operation down to DC is possible. Under software control the ST72E71,T71 can be placed in WAIT or STOP mode thus reducing power consumption. The enhanced instruction set and addressing modes afford real programming potential.

The device includes an on-chip oscillator, CPU, ROM, RAM, EEPROM, I/O, a timer with 2 input capture and 2 output compare signals, an 8-channel Analog to Digital Converter and an industry standard SPI as standard peripherals.

Dedicated functions include a Sync Processor for video timing analysis, East-West Pin Cushion automatic correction and 18 PWM/BRM outputs for analog control of external functions.

Figure 2. ST7271 Block Diagram



Note 1 : ROM is replaced by EPROM for EPROM/OTP versions.

4.2 PIN DESCRIPTION

V_{DD}. Power supply voltage

V_{SS}. Digital and Analog Ground

V_{DDA}. Analog V_{DD} and reference for EWPCC Digital to Analog Converter (DAC, 8 Volts).

V_{SSA}. Analog V_{SS} for EWPCC DAC.

OSCIN, OSCOUT. Oscillator input and output pins. These pins are to be connected to a parallel resonant crystal or ceramic resonator. An external clock source can also be connect to OSCIN.

$\overline{\text{RESET}}$. The active low input signal forces the initialization of the MCU. This event is the top priority non maskable interrupt. This pin is switched low when the Watchdog has triggered. It can be used to reset external peripherals.

TEST/V_{PP}. This pin must be held low for normal application. In the EPROM programming mode, this pin acts as the programming voltage input V_{PP}.

VFBACK (PB0). Vertical Flyback signal (TTL level). This pin accepts the Vertical Flyback signal used for timing correlation for the East-west Pin Cushion correction when this is used or is PB0.

EWPCC. Analog output of correction signal from East-West Pin Cushion controller (2-6V, I_{OUT} = 1mA).

OCMP (PC0). Output compare signal coming from the TIMER. This output signal, according to a register bit option, can be the OCMP pin (for output compare 1 of the timer) or the PC0 pin.

MISO (PC2). SPI Master Out/Slave In Data Output/Input when SPI is enabled or PC2.

MOSI (PC3). SPI Master In/Slave Out Data Input/Output when SPI is enabled or PC3.

SCK (PC4). SPI Serial Clock when SPI is enabled or PC4

$\overline{\text{SS}}$ (PC5). SPI Slave Select when SPI is enabled or PC5.

VSYNCI. Vertical Sychronization Input (TTL level)

HSYNCI. Horizontal Sychronization Input (TTL level)

CSYNCI (PD0). Composite Sychronization Input (TTL level). This pin accepts the composite synchronisation input when the Sync Processor I/O functions are enabled or is PD0.

HSYNCO (PD1). Horizontal Sychronization Output. This pin outputs the horizontal synchronisation output from the Sync Processor (or HSYNCI) when the Sync Processor I/O functions are enabled or is PD1.

VSYNCO (PD2). Vertical Sychronization Output. This pin outputs the vertical synchronisation output from the Sync Processor (or VSYNCI) when the Sync Processor I/O functions are enabled or is PD2.

CLMPO (PD3). Clamp Output. This pin outputs the clamping (back porch) output signal from the Sync Processor (or HSYNCI) when the Sync Processor I/O functions are enabled or is PD3.

DA2-DA17 (56-pin package),
DA2-DA9 (42-pin package), 10-bit PWM/BRM outputs (for Analog controls, after external filtering)

DA0, DA1. 12-bit PWM/BRM outputs (for Analog Controls, after external filtering).

PA0-PA7, PB0-PB7, PC0-PC5, PD0-PD4 (56 pin package). These 27 lines are standard I/O lines, programmable as either input or output.

- **PORT A.** 8 I/O lines, bit programmable, accessed through DDRA and DRA Registers. Each bit can be defined as a standard input port bit without pull-up resistor or as an open drain output port (up to 12V).

- **PORT B**. 8 Standard I/O lines bit programmable accessed through DDRB and DRB Registers. Each bit can be programmed as an analog input (by control bits in the PORT B Configuration register), digital input (with internal pull-up resistor), push-pull digital output or as interrupt wake-up (with pull-up). These negative edge or low-level sensitive interrupt lines can wake-up the ST7271 from WAIT or STOP mode. This feature allows to build low power applications when the ST7271 can be waken-up from keyboard push.
PB0 is used for the East-West Pin cushion controller VFBACK input as shown above when the EWPCC is used.

- **PORT C.** 6 Standard I/O lines accessed through DDRC and DRC Registers. Each bit can be programmed as digital input (with or without pull-up internal resistor), open drain output or SPI control and data signals (as shown for the dedicated SPI signals above) whenever the SPI is active, the outputs are in the pull-pull configuration.
The pull-up resistor is enabled for all bits present by one control bit in the Programmable Input/Output Configuration Register. The resistor is automatically disabled for the pins used for the SPI when the SPI is enabled.

- **PORT D.** 4 Standard I/O lines bit programmable accessed through DDRD and DRD Registers. Each bit can be programmed as an input (with internal pull-up resistor), push-pull output or Synchronization inputs and outputs to/from the Sync Processor. When programmed as inputs, Video Synchronisation signals can be directly inspected. The inputs may also be passed through the Sync Processor to the Timer Input Captures

These pin functions are also summarised in the following table, which also indicates the availability of functions for the 42-pin SDIP package.

PIN DESCRIPTION (Continued)

Table 1. ST72E71,T71 Pin Description

Pin Name	Pin Function(s)	56 Pins	42 Pins
V_{DDA}	Analog V_{DD} for EWPCC	1	1
EWPCC	EWPCC output voltage	2	2
DA0	12-bit PWM/BRM output*	3	3
DA1	12-bit PWM/BRM output	4	4
DA2	10-bit PWM/BRM output*	5	5
DA3	10-bit PWM/BRM output	6	6
DA4	10-bit PWM/BRM output	7	7
DA5	10-bit PWM/BRM output	8	8
DA6	10-bit PWM/BRM output	9	9
DA7	10-bit PWM/BRM output	10	10
DA8	10-bit PWM/BRM output	11	11
DA9	10-bit PWM/BRM output	12	12
PB7	I/O Port PB7	13	
PB6	I/O Port PB6	14	
PB5	I/O Port PB5	15	
PB4	I/O Port PB4	16	
PB3	I/O Port PB3	17	13
PB2	I/O Port PB2	18	14
PB1	I/O Port PB1	19	15
VFBACK/PB0	I/O Port PB0 /VFBACK Input	20	16
PD4	I/O Port PD4	21	
CLMPO/PD3	I/O Port PD3/Clamp Output	22	17
DA10	10-bit PWM/BRM output 10	23	
DA11	10-bit PWM/BRM output 11	24	
$\overline{\text{RESET}}$	Reset Input/Output	25	18
VSYNCO/PD2	I/O Port PD2/VSYNC Output	26	19
VSYNCI	VSYNC Input to Sync Processor	27	20
V_{DD}	Power Supply	28	21
HSYNCI	HSYNC Input to Sync Processor	29	22
HSYNCO/PD1	I/O Port PD1/HSYNC Output	30	23
CSYNCI/PD0	I/O Port PD0/CSYNC Input	31	24
OSCOUT	Oscillator Output	32	25
OSCIN	Oscillator Input	33	26
DA12	10-bit PWM/BRM output 12	34	
DA13	10-bit PWM/BRM output 13	35	
PA7	I/O Port PA7	36	27
PA6	I/O Port PA6	37	28
PA5	I/O Port PA5	38	29
PA4	I/O Port PA4	39	30
PA3	I/O Port PA3	40	31
PA2	I/O Port PA2	41	32
PA1	I/O Port PA1	42	33
PA0	I/O Port PA0	43	34
DA14	10-bit PWM/BRM output 14	44	
DA15	10-bit PWM/BRM output 15	45	
DA16	10-bit PWM/BRM output 16	46	
DA17	10-bit PWM/BRM output 17	47	
TEST/V_{PP}	TEST pin must be held low. EPROM Prog. voltage input	48	35
PC0/OCMP	I/O Port PC0, Timer Output Compare	49	36
PC1	I/O Port PC1	50	
PC2/MISO	I/O Port PC2, SPI Data	51	37
PC3/MOSI	I/O Port PC3, SPI Data	52	38
PC4/SCK	I/O Port PC4, SPI Clock output	53	39
PC5/$\overline{\text{SS}}$	I/O Port PC5, SPI Slave Select	54	40
V_{SS}	Digital ground	55	41
V_{SSA}	Analog ground for EWPCC	56	42

Note * : Open Drain

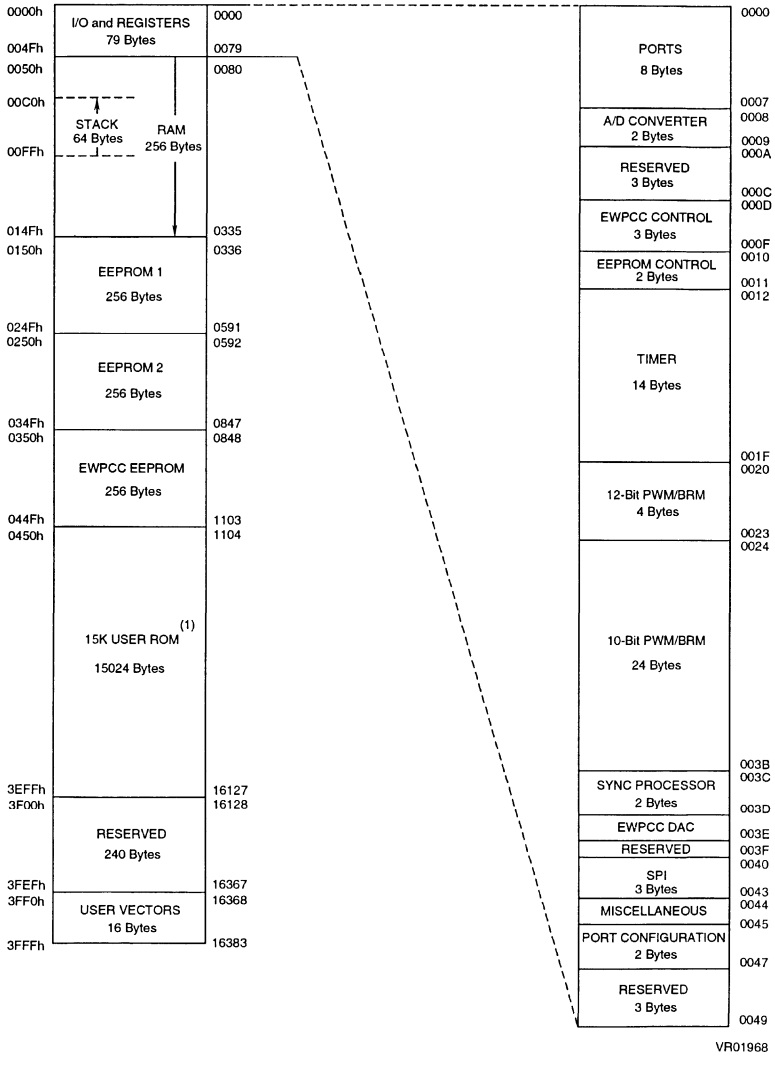
4.3 MEMORY MAP

As shown in Figure 3, the MCU is capable of addressing 16K bytes of memory and I/O registers. In the ST72E71,T71, 16383 of these bytes are user accessible.

The available memory locations consist of 80 bytes of I/O registers, 256 bytes of RAM, 512 bytes of EEPROM, 256 bytes of EWPCC EEPROM and 15Kbytes of user EPROM. The RAM space includes 64 bytes for the stack from 014Fh to 010Fh.

The highest address bytes contain the user defined reset and interrupt vectors.

Figure 3. Memory Map



Note 1 : ROM is replaced by EPROM for EPROM/OTP versions.

4.4 I/O and REGISTER MAP

Address	Register Name
0000h	PORT A DATA REGISTER
0001h	PORT B DATA REGISTER
0002h	PORT C DATA REGISTER
0003h	PORT D DATA REGISTER
0004h	PORT A DATA DIRECTION REGISTER
0005h	PORT B DATA DIRECTION REGISTER
0006h	PORT C DATA DIRECTION REGISTER
0007h	PORT D DATA DIRECTION REGISTER
0008h	A/D DATA REGISTER
0009h	A/D CONTROL/STATUS REGISTER
000Ah-000Ch	Reserved
000Dh	EWPCC0 REGISTER
000Eh	EWPCC1 REGISTER
000Fh	EWPCC EEPROM CONTROL REGISTER
0010h	EEPROM 1 CONTROL REGISTER
0011h	EEPROM 2 CONTROL REGISTER
0012h	TIMER CONTROL REGISTER
0013h	TIMER STATUS REGISTER
0014h	INPUT CAPTURE REGISTER 1, High
0015h	INPUT CAPTURE REGISTER 1, Low
0016h	OUTPUT COMPARE REGISTER 1, High
0017h	OUTPUT COMPARE REGISTER 1, Low
0018h	COUNTER REGISTER, High
0019h	COUNTER REGISTER, Low
001Ah	ALTERNATE COUNTER REGISTER, High
001Bh	ALTERNATE COUNTER REGISTER, Low
001Ch	INPUT CAPTURE REGISTER 2, High
001Dh	INPUT CAPTURE REGISTER 2, Low
001Eh	OUTPUT COMPARE REGISTER 2, High
001Fh	OUTPUT COMPARE REGISTER 2, Low
0020h	PWM0 - 12 bit PWM/BRM
0021h	BRM0 - 12 bit PWM/BRM
0022h	PWM1 - 12 bit PWM/BRM
0023h	BRM1 - 12 bit PWM/BRM
0024h	PWM2 - 10 bit PWM/BRM
0025h	BRM2+BRM3

Address	Register Name
0026h	PWM3
0027h	PWM4
0028h	BRM4+BRM5
0029h	PWM5
002Ah	PWM6
002Bh	BRM6+BRM7
002Ch	PWM7
002Dh	PWM8
002Eh	BRM8+BRM9
002Fh	PWM9
0030h	PWM10
0031h	BRM10+BRM11
0032h	PWM11
0033h	PWM12
0034h	BRM12+BRM13
0035h	PWM13
0036h	PWM14
0037h	BRM14+BRM15
0038h	PWM15
0039h	PWM16
003Ah	BRM16+BRM17
003Bh	PWM17
003Ch	SYNC MUX CONTROL REGISTER
003Dh	SYNC COUNTER CONTROL REGISTER
003Eh	EWPCC DAC REGISTER
003Fh	Reserved
0040h	SPI DATA I/O REGISTER
0041h	Reserved
0042h	SPI CONTROL REGISTER
0043h	SPI STATUS REGISTER
0044h	MISCELLANEOUS REGISTER
0045h	PORT B CONFIGURATION REGISTER
0046h	PROGRAMMABLE INPUT/OUTPUT CONFIGURATION REGISTER
0047h	Reserved
0048h	Reserved
0049h	Reserved

4.5 ST72E71,T71 EPROM/OTP DESCRIPTION

The ST72E71 is the EPROM version of the ST7271 ROM product. The ST72T71 OTP has the same characteristics. Both include EPROM memory instead of the ROM memory of the ST7271, and so the program and constants of the program can be easily modified by the user with the ST72E71 EPROM Programming Board from SGS-THOMSON.

From a user point of view the products have exactly the same software and hardware features of the ROM version. An additional mode is used to configure the part for programming of the EPROM, this is set by a +12.5V voltage applied to the TEST/V_{PP} pin. The programming is described in the User Manual of the EPROM Programming board.

Other than this additional mode, the ST72E71,T71 parts are fully compatible with the ROM ST7271 equivalent, this datasheet thus provides only information specific to the EPROM based devices.

THE READER IS ASKED TO REFER TO THE DATASHEET OF THE ST7271 ROM-BASED DEVICE FOR FURTHER DETAILS.

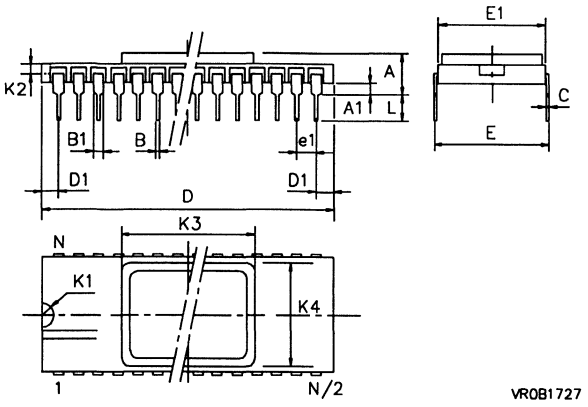
4.5.1 EPROM ERASING

The EPROM of the windowed package of the ST72E71 may be erased by exposure to Ultra Violet light.

The erasure characteristic of the ST72E71 EPROM is such that erasure begins when the memory is exposed to light with wave lengths shorter than approximately 4000Å. It should be noted that sunlight and some types of fluorescent lamps have wavelengths in the range 3000-4000Å. It is thus recommended that the window of the ST72E71 package be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.
The recommended erasure procedure of the ST72E71 EPROM is exposure to short wave ultraviolet light which has wavelength 2537Å. The integrated dose (i.e. UV intensity x exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm^2 power rating. The ST72E71 should be placed within 2.5 cm (1 inch) of the lamp tubes during erasure.

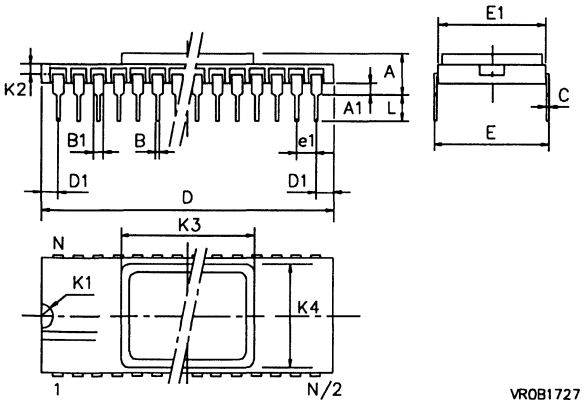
4.6 PACKAGE MECHANICAL DATA

56-Pin Ceramic Shrink Dual-In-line Package, 600 Mil Width with Window



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D						
D1	–	–	–	–	–	–
E	15.2 BSD			0.600 BSD		
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	56					

42-Pin Ceramic Shrink Dual-In-line Package, 600 Mil Width with Window



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D	36.32		39.12	1.430		1.540
D1	–	–	–	–	–	–
E	15.2 BSD			0.600 BSD		
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	42					

ST72E71/T71 MICROCONTROLLER OPTION LIST

Customer .....
Address: .....
Contact: .....
Phone No: .....
Reference: .....

SGS-THOMSON Microelectronics references

Device:	EPROM	OTP
SDIP56	[ ] ST72E71N5	[ ] ST72T71N5
SDIP42	[ ] ST72E71J1	[ ] ST72T71J1

Temperature Range [ ] 0 to 70°C

OPTION LIST:

Watchdog State After Reset	[ ] Enable	[ ] Disable
Watchdog during WAIT mode	[ ] Active	[ ] Suspend
Input Clock to SPI (8MHz osc)	[ ] 2MHz	[ ] 4MHz

Signature .....

Date .....

4.7 ORDERING INFORMATION

Ordering Information Table

Sales Type	EPROM (Bytes)	OTP ROM (Bytes)	RAM (Bytes)	EEPROM (Bytes)	Temperature Range	Package
ST727E1N5D1 ST727E1J1D1	16K 8K		256 192	512 384	-0 to + 70°C	CSDIP56W CSDIP42W
ST727T1N5B1 ST727T1J1B1		16K 8K	256 192	512 384	-0 to + 70°C	PSDIP56 PSDIP42

8/16 Bit MCUs

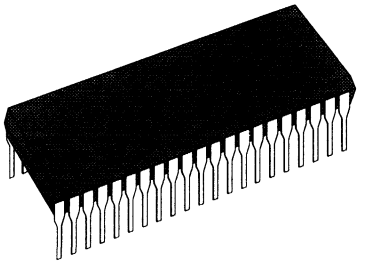
SGS-THOMSON MICROELECTRONICS

ST9291

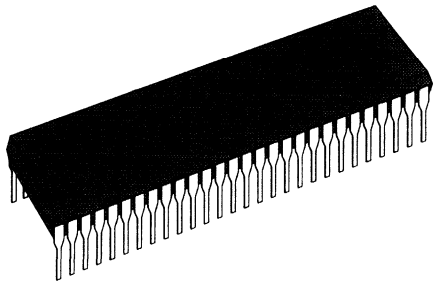
16-32K ROM HCMOS MCU WITH ON SCREEN DISPLAY AND VOLTAGE TUNING OUTPUT

PRELIMINARY DATA

- Register oriented 8/16 bit CORE with RUN, WFI and HALT modes
- Minimum instruction cycle time: 500ns (12MHz internal)
- 16 to 32K bytes of ROM, 384/640 bytes of RAM, 224 general purpose registers available as RAM, accumulators or index registers (Register File)
- 42-lead Shrink DIP package or 56-lead Shrink DIP package
- Interrupt handler and Serial Peripheral Interface as standard features
- 31 (42 pin package) / 42 (56 pin package) fully programmable I/O pins
- 34 character x15 rows software programmable On Screen Display module with colour, italic, underline, flash, transparent and fringe attribute options
- 14-bit Voltage Synthesis for tuning reference voltage.
- 8 8-bit PWM D/A outputs with repetition frequency 2 to 32kHz and 12V Open Drain Capability
- 16 bit Timer with 8 bit Prescaler, able to be used as a Watchdog Timer
- 16-bit programmable Slice Timer with 8-bit prescaler
- 3 channel Analog to Digital Converter, with integral sample and hold, fast 5.75µs conversion time, 6-bit guaranteed resolution
- Rich Instruction Set and 14 Addressing modes
- Division-by-Zero trap generation
- Versatile Development tools, including assembler, linker, C-compiler, archiver, graphic oriented debugger and hardware emulators
- Real Time Operating System
- Windowed EPROM parts available for prototyping and pre-production development phases



PSDIP42



PSDIP56

(Ordering Information at the end of the Datasheet)

Figure 1. 42 Pin Shrink DIP Pinout

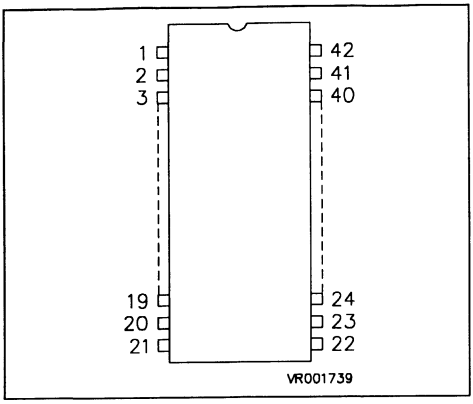
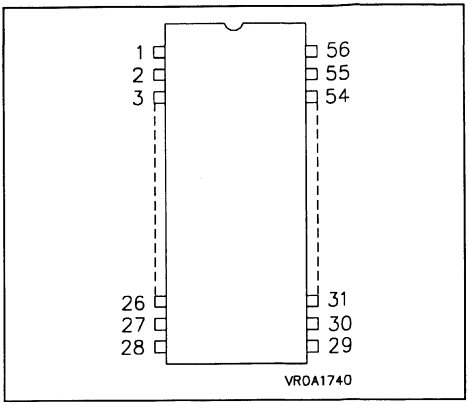


Figure 2. 56 Pin Shrink DIP Pinout



ST9291J Pin Description

Pin	Pin name	Pin	Pin name
1	P2.0/INT7	42	P2.1/INT5/AIN1
2	$\overline{\text{RESET}}$	41	P2.2/INT0/AIN2
3	P0.7	40	P2.3/INT6/VSO1
4	P0.6	39	P2.4/NMI
5	P0.5	38	P2.5/AIN3/VSO2
6	P0.4	37	OSCIN
7	P0.3	36	OSCOUT
8	P0.2	35	P4.7/PWM7/ EXTRG (AD)
9	P0.1	34	P4.6/PWM6
10	P0.0	33	P4.5/PWM5
11	P3.7	32	P4.4/PWM4
12	P3.6	31	P4.3/PWM3
13	P3.5	30	P4.2/PWM2
14	P3.4	29	P4.1/PWM1
15	P3.3/B	28	P4.0/PWM0
16	P3.2/G	27	VSYNC
17	P3.1/R	26	HSYNC
18	P3.0/FB	25	AV_{DD}
19	P5.1/SDIO	24	PLLR
20	P5.0/SCK/INT2	23	PLLF
21	V_{DD}	22	V_{SS}

Notes (N Package only) :

1. N.C. means "not connected"
2. Pins 14 and 28 (V_{DD}) are internally connected

ST9291N Pin Description

Pin	Pin name	Pin	Pin name
1	P2.1/INT5/AIN1	56	P2.2/INT0/AIN2
2	P2.0/INT7	55	P2.3/INT6/VSO1
3	$\overline{\text{RESET}}$	54	P2.4/NMI
4	P0.7	53	P2.5/AIN3/VSO2
5	P0.6	52	P1.0
6	P0.5	51	P1.1
7	N.C.(1)	50	P1.2
8	P0.4	49	P1.3
9	P0.3	48	P1.4
10	P0.2	47	P1.5
11	P0.1	46	P1.6
12	P0.0	45	P1.7
13	N.C.(1)	44	OSCIN
14	V_{DD}(2)	43	OSCOUT
15	N.C.(1)	42	P4.7/PWM7/ EXTRG (AD)
16	P3.7	41	P4.6/PWM6
17	P3.6	40	P4.5/PWM5
18	P3.5	39	P4.4/PWM4
19	P3.4	38	P4.3/PWM3
20	P3.3/B	37	P4.2/PWM2
21	P3.2/G	36	P4.1/PWM1
22	P3.1/R	35	P4.0/PWM0
23	P3.0/FB	34	VSYNC
24	P5.3	33	HSYNC
25	P5.2	32	AV_{DD}
26	P5.1/SDIO	31	PLLR
27	P5.0/SCK/INT2	30	PLLF
28	V_{DD}(2)	29	V_{SS}

GENERAL DESCRIPTION

The ST9291 is a ROM member of the ST9 family of microcontrollers, completely developed and produced by SGS-THOMSON Microelectronics using a proprietary n-well HCMOS process.

The ROM parts are fully compatible with their EPROM versions, which may be used for the prototyping and pre-production phases of development.

The nucleus of the ST9291 is the advanced Core which includes the Central Processing Unit (CPU), the Register File, a 16-bit Timer/Watchdog with 8-bit Prescaler, a Serial Peripheral Interface supporting S-bus, I^2C-bus and IM-bus Interface, plus two 8-bit I/O ports. The Core has independent memory and register buses allowing a high degree of pipelining to add to the efficiency of the code execution speed of the extensive instruction set.

The powerful I/O capabilities demanded by microcontroller applications are fulfilled by the ST9291 with up to 32/42 I/O lines dedicated to digital Input/Output. These lines are grouped into up to six I/O Ports and can be configured on a bit basis under software control to provide timing, status signals, timer inputs and outputs, analog inputs, external interrupts and serial or parallel I/O.

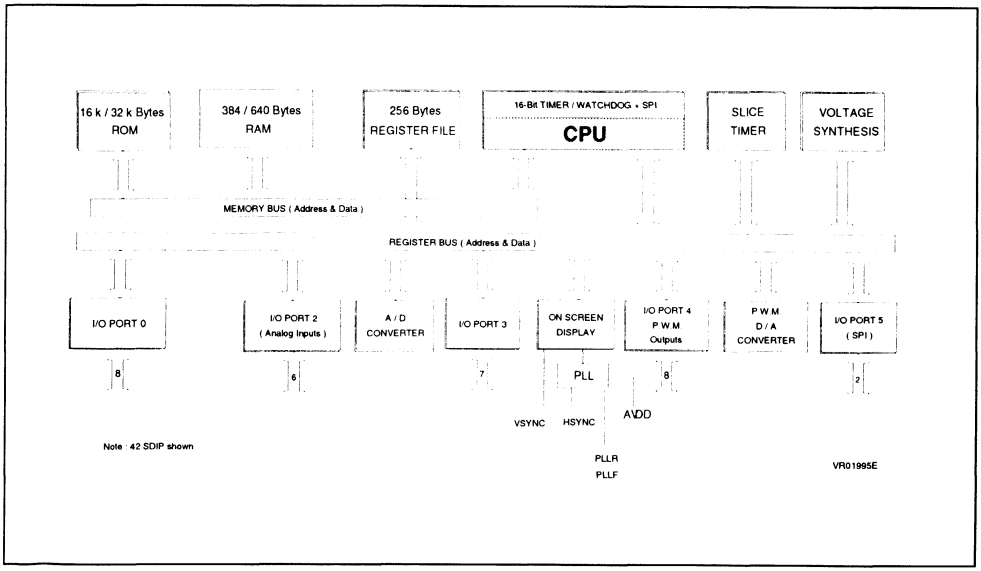
Three basic memory spaces are available to support this wide range of configurations: Program Memory, Data Memory and the Register File, which includes the control and status registers of the on-chip peripherals.

The 16-bit Slice Timer with an 8-bit Prescaler.

The human interface is provided by the On Screen Display module, this can produce up to 15 lines of up to 34 characters from a ROM defined 128 character set. The 9x13 character can be modified by 4 different pixel sizes, with character rounding, and formed into words with colour and format attributes.

14-bit Voltage Synthesis using the PWM (Pulse Width Modulation)/BRM (Bit Rate Modulation) technique to generate tuning voltages for low-mid range TV set applications. The tuning voltage is output on one of two separate output pins.

Figure 3. ST9291 Block Diagram



GENERAL DESCRIPTION (Continued)

Control of TV settings is able to be made with up to eight 8-bit PWM outputs, with a frequency maximum of 23,437Hz at 8-bit resolution (INTCLK = 12MHz). Low resolutions with higher frequency operation can be programmed.

In addition there is a 3 channel Analog to Digital Converter with integral sample and hold, fast 5.75µs conversion time and 6-bit guaranteed resolution.

PIN DESCRIPTION

VSYNC. *Vertical Synch.* Vertical video synchronisation input to OSD. Positive or negative polarity.

HSYNC. *Horizontal Synch.* Horizontal video synchronisation input to OSD. Positive or negative polarity.

PLLF. *PLL Filter input.* Filter input for the OSD for PLL feed-back.

PLLR. *PLL Resistor connection pin.* For resistor connection to select the PLL gain adjust.

$\overline{\text{RESET}}$. *Reset (input, active low).* The ST9 is initialised by the Reset signal. With the deactivation of $\overline{\text{RESET}}$, program execution begins from the Program memory location pointed to by the vector contained in program memory locations 00h and 01h.

OSCIN, OSCOUT. *Oscillator (input and output).* These pins connect a parallel-resonant crystal (24MHz maximum), or an external source to the on-chip clock oscillator and buffer. OSCIN is the input of the oscillator inverter and internal clock generator; OSCOUT is the output of the oscillator inverter.

AV_{DD}. Analog V_{DD} of PLL. This pin must be tied to V_{DD} externally to the ST9291.

V_{DD}. Main Power Supply Voltage (5V10%)

V_{SS}. Digital Circuit Ground.

P0.0-P0.7, P2.0-P2.5, P3.0-P3.7, P4.0-P4.7, P5.0-P5.1 (J suffix)
P0.0-P0.7, P1.0-P1.7, P2.0-P2.5, P3.0-P3.7, P4.0-P4.7, P5.0-P5.3 (N suffix) *I/O Port Lines (Input/Output, TTL or CMOS compatible).* 32/42 lines grouped into I/O ports, bit programmable under program control as general purpose I/O or as Alternate functions (see next section).

P4.0 - P4.7 are high voltage (12V) open drain outputs. The voltage in open drain output mode for all other I/O bits must not exceed V_{DD}.

I/O Port Alternate Functions.

Each pin of the I/O ports of the ST9291 may assume software programmable Alternative Functions as shown in the Pin Configuration Drawings. Table 1 shows the Functions allocated to each I/O Port pin.

PIN DESCRIPTION (Continued)

Table 1.ST9291 I/O Port Alternative Function Summary

I/O PORT	Name	Function	Alternate Function	Pin Assignment	
Port.bit				9291J	9291N
P0.0		I/O		10	12
P0.1		I/O		9	11
P0.2		I/O		8	10
P0.3		I/O		7	9
P0.4		I/O		6	8
P0.5		I/O		5	6
P0.6		I/O		4	5
P0.7		I/O		3	4
P1.0		I/O		-	52
P1.1		I/O		-	51
P1.2		I/O		-	50
P1.3		I/O		-	49
P1.4		I/O		-	48
P1.5		I/O		-	47
P1.6		I/O		-	46
P1.7		I/O		-	45
P2.0	INT7	I	External Interrupt 7 with Schmitt Trigger	1	2
P2.1	INT5	I	External Interrupt 5 with Schmitt Trigger	42	1
P2.1	AIN1	I	A/D Analog Input 1	42	1
P2.2	INT0	I	External Interrupt 0	41	56
P2.2	AIN2	I	A/D Analog Input 2	41	56
P2.3	INT6	I	External Interrupt 6	40	55
P2.3	VSO1	O	Voltage Synthesis Output 1	40	55
P2.4	NMI	I	Non-Maskable Interrupt	39	54
P2.5	AIN3	I	A/D Analog Input 3	38	53
P2.5	VSO2	O	Voltage Synthesis Output 2	38	53
P3.0	FB	O	Fast Blanking OSD output	18	23
P3.1	R	O	Red Video Colour OSD output	17	22
P3.2	G	O	Green Video Colour OSD output	16	21
P3.3	B	O	Blue Video Colour OSD output	15	20

PIN DESCRIPTION (Continued)

Table 1. ST9291 I/O Port Alternative Function Summary (Continued)

I/O PORT	Name	Function	Alternate Function	Pin Assignment	
Port.bit				9291J	9291N
P3.4		I/O		14	19
P3.5		I/O		13	18
P3.6		I/O		12	17
P3.7		I/O		-	16
P4.0	PWM0	O	PWM Output 0	28	35
P4.1	PWM1	O	PWM Output 1	29	36
P4.2	PWM2	O	PWM Output 2	30	37
P4.3	PWM3	O	PWM Output 3	31	38
P4.4	PWM4	O	PWM Output 4	32	39
P4.5	PWM5	O	PWM Output 5	33	40
P4.6	PWM6	O	PWM Output 6	34	41
P4.7	PWM7	O	PWM Output 7	35	42
P4.7	EXTRG	I	A/D External Trigger	35	42
P5.0	SCK	O	SPI Serial Clock (1)	20	27
P5.0	INT2	I	External Interrupt 2 (1)	20	27
P5.1	SDIO	I/O	SPI Serial Data Input/Output (1)	19	26
P5.2		I/O		-	25
P5.3		I/O		-	24

Notes.

1. The alternate functions of SCK/INT2 and SDIO may be swapped by using the SWAP Register Function.
2. Schmitt trigger options are available as a mask option for any input pin.

1 CORE DESCRIPTION

1.1 CORE ARCHITECTURE

1.1.1 INTRODUCTION

The Core or Central Processing Unit (CPU) of the ST9 includes the 8 bit Arithmetic Logic Unit and the 16 bit Program Counter, System and User Stack Pointers. The microcoded Instruction Set is highly optimised for both byte (8 bit) and word (16 bit) data, BCD and Boolean data types, with 14 addressing modes.

Three independent buses are controlled by the Core, a 16 bit Memory bus, an 8 bit Register addressing bus and a 6 bit Interrupt/DMA bus connected to the interrupt and DMA controllers in the on-chip peripherals and the Core. This multiple bus architecture allows a high degree of pipelining and parallel operation, giving the ST9 its efficiency in both numerical calculations and communication with the on-chip peripherals.

1.1.2 ADDRESS SPACES

The ST9 has three separate address spaces:

- Register File: 240 8-bit registers plus up to 64 pages of 16 bytes each, located in the on-chip peripherals.
- Data memory with up to 64K (65536) bytes
- Program memory with up to 64K (65536) bytes

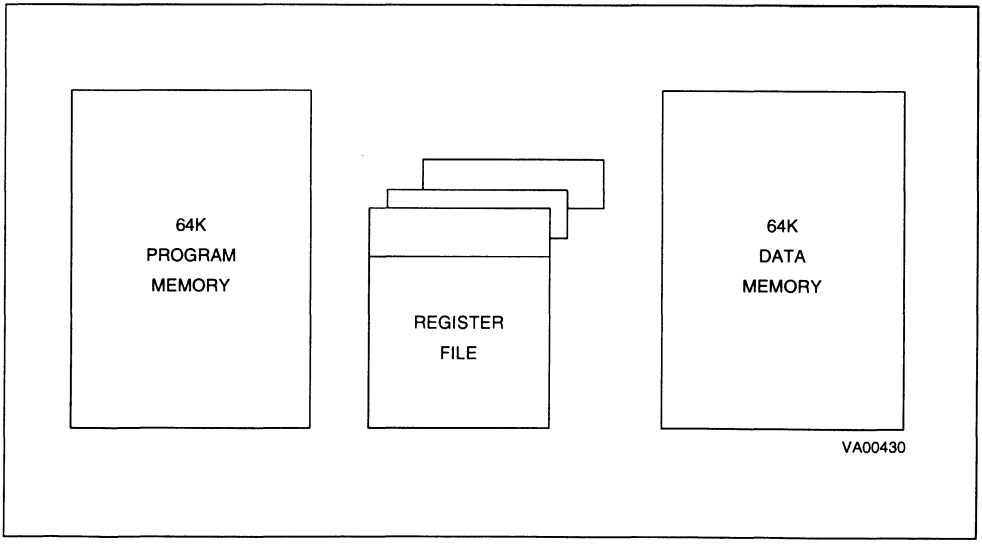
The Data and Program memory spaces will be addressed in further detail in the next section.

1.1.2.1 Register File

The Register File consists of:

- 224 general purpose registers R0 to R223
- 16 system registers in the System Group (R224 to R239).
- I/O pages depending on the configuration of the ST9, each containing up to 16 registers, with paging facilities based on the top group (R240 to R255).

Figure 1-4. Address Spaces



ADDRESS SPACES (Continued)

Figure 1-5. Register Grouping

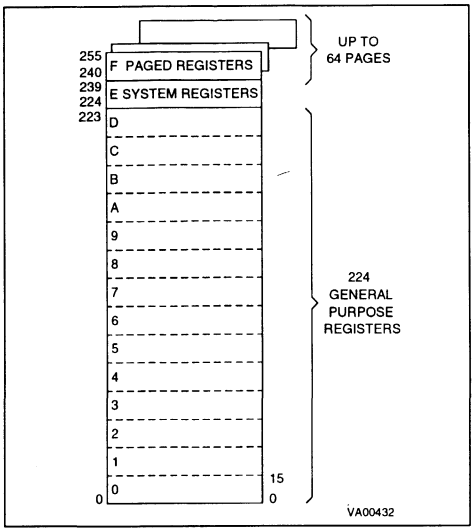


Figure 1-6. Page Pointer Configuration

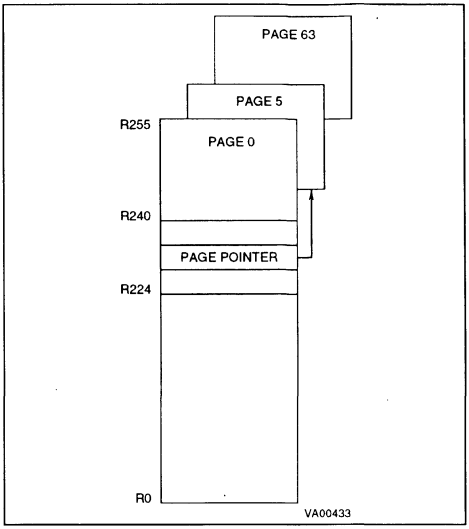
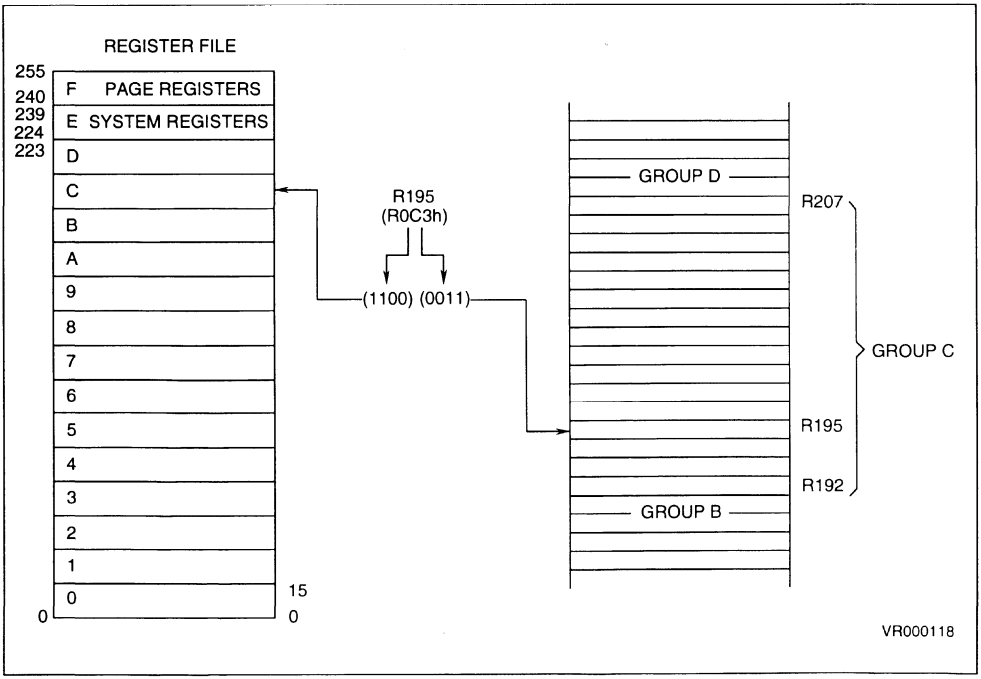


Figure 1-7. Addressing the Register File



ADDRESS SPACES (Continued)

1.1.2.2 Addressing Registers

All registers in the Register File and pages can be specified by using a decimal, hex or binary address, e.g. R231, RE7h or R11100111b is the same register.
The registers can be referred to by their hexadecimal group address, so that registers R0-R15 form group 0, R160-R175 form group A and so on.

Working Register Addresses

The 8-bit register address is formed by 2 nibbles, for example, for register R195 or RC3h or R11000011, 1100 specifies the 13th group (i.e. group C) and 0011 specifies the 3rd register in that group.

Working registers are addressed by supplying the least significant nibble in the instruction and adding it to the most significant nibble found in the Register Pointer (R233). Working register addressing is shown in Figure 1-7.

System Registers

The 16 system registers at addresses R224 to R239 form Group E.
The system registers are addressable using any of the 4 register addressing modes and the most significant nibble will, in all cases, be 14 (0Eh).

Paged Registers

There are a maximum of 64 pages each containing 16 registers. These are addressed using the register addressing modes with the addition of the Page Pointer register, R234. This register selects the page to be addressed in group F and once set, does not need to be changed if two or more registers on the same page are to be addressed in succession.

Therefore if the Page Pointer, R234, is set to 5, the instructions

```
spp 5
ld R242, r4
```

will load the contents of working register r4 into the third register (R242) of page 5.

These paged registers hold data and control registers related to the on-chip peripherals, and thus the configuration depends upon the peripheral organisation of each ST9 family member. i.e. pages only exist if the peripheral exists.

Available pages are shown in Table 1-3.

1.1.2.3 Input/Output Ports

The Input/Output ports are located in two areas. The port registers for Ports 0-5 are located at the bottom of the System register group in locations R224 to R229.

Each Port has three associated Control registers, which determine the individual pin modes (I/O, Open-Drain etc). These registers are located in pages 2 and 3.

Table 1-2. Register File Organization

Hex. Address	Decimal Address	Function	Register File Group
F0-FF	240-255	Paged Regicters	Group F
E0-EF	224-239	System Registers	Group E
D0-DF	208-223	General Purpose Registers	Group D
C0-CF	192-207		Group C
B0-BF	176-191		Group B
A0-AF	160-175		Group A
90-9F	144-159		Group 9
80-8F	128-143		Group 8
70-7F	112-127		Group 7
60-6F	96-111		Group 6
50-5F	80-95		Group 5
40-4F	64-79		Group 4
30-3F	48-63		Group 3
20-2F	32-47		Group 2
10-1F	16-31		Group 1
00-0F	00-15		Group 0

ADDRESS SPACES (Continued)

Table 1-3. Group F Peripheral Organization

Applicable for ST9291

	HEX	00	02	03	0B	28	29	2A	3B	3E	
DEC		00	02	03	11	40	41	42	59	62	
R255	RFF	SWAP	RESER						VSO		RFF
R254	RFE	SPI						RESER			RFE
R253	RFD		PORT 3						RESER		RFD
R252	RFC	WCR									RFC
R251	RFB		RESER	RESER							RFB
R250	RFA	T/WD			RESER						RFA
R249	RF9		PORT 2			OSD	OSD			RESER	RF9
R248	RF8					CHAR	CHAR	OSD			RF8
R247	RF7		RESER			1 to 16	17 to 32		PWM		RF7
R246	RF6										RF6
R245	RF5	EXT INT	PORT1	PORT 5							RF5
R244	RF4										RF4
R243	RF3		RESER	RESER				OSD			RF3
R242	RF2				SLICE			CHAR			RF2
R241	RF1	RESER	PORT 0	PORT 4	TIMER			33 to 36		A/D	RF1
R240	RF0									CONV	RF0

1.1.3 SYSTEM REGISTERS

Following is the description of System Registers. For PORT0 to PORT5 Registers, please refer to I/O Port Chapter.

Figure 1-8. System Register

R239 (EFh)	SYS. STACK POINTER LOW
R238 (EEh)	SYS. STACK POINTER HIGH
R237 (EDh)	USER STACK POINTER LOW
R236 (ECh)	USER STACK POINTER HIGH
R235 (EBh)	MODE REGISTER
R234 (EAh)	PAGE POINTER
R233 (E9h)	REGISTER POINTER 1
R232 (E8h)	REGISTER POINTER 0
R231 (E7h)	FLAGS
R230 (E6h)	CENTRAL INT. CNTL REG
R229 (E5h)	PORT5
R228 (E4h)	PORT4
R227 (E3h)	PORT3
R226 (E2h)	PORT2
R225 (E1h)	PORT1
R224 (E0h)	PORT0

1.1.3.1 Central Interrupt Control Register

This Register CICR is located in the system Register Group at the address R230 (E6h). Please refer to "INTERRUPT" and "DMA" chapters in order to get the background of the ST9 interrupt philosophy.

CICR R230 (E6h) System Read/Write
Central Interrupt Control Register

Reset Value : 1000 0111

7							0
GCEN	TLIP	TLI	IEN	IAM	CPL2	CPL1	CPL0

b7 = **GCEN:** *Global Counter Enable.* This bit is the Global Counter Enable of the Multifunction Timers. The GCEN bit is ANDed with the CE (Counter Enable) bit of the Timer Control Register (explained in the Timer chapter) in order to enable the Timers when both bits are set. This bit is set after the Reset cycle.

b6 = **TLIP:** *Top Level Interrupt Pending.* This bit is automatically set when a Top Level Interrupt Request is recognized. This bit can also be set by Software in order to simulate a Top Level Interrupt Request.

b5 = **TLI:** *Top Level Interrrupt bit.* When this bit is set, a Top Level interrupt request is acknowledged depending on the IEN bit and the TLNM bit (in Nested Interrupt Control Register). If the TLM bit is reset the top level interrupt acknowledgement depends on the TLNM alone.

b4 = **IEN:** *Enable Interrupt.* This bit, (when set), allows interrupts to be accepted. When reset no interrupts other than the NMI can be acknowledged. It is cleared by interrupt acknowledgement for concurrent mode and set by interrupt return (`iret`). It can be managed by hardware and software (`ei` and `di` instruction).

b3 = **IAM:** *Interrupt Arbitration Mode.* This bit covers the selection of the two arbitration modes, the Concurrent Mode being indicated by the value "0" and the Fully Automatic Nested Mode by the value "1". This bit is under software control.

b2-b0 = **CPL2-CPL0:** *Current Priority Level.* These three bits record the priority level of the interrupt presently under service (i.e. the Current Priority Level, CPL). For these priority levels 000 is the highest priority and 111 is the lowest priority. The CPL bits can be set by hardware or software and give the reference by which following interrupts are either left pending or able to interrupt the current interrupt. When the present interrupt is replaced by one of a greater priority, the current priority value is automatically stored until required.

SYSTEM REGISTERS (Continued)

1.1.3.2 Flag Register

The Flag Register contains 8 flags indicating the status of the ST9. During an interrupt the flag register is automatically stored in the system stack area and recalled at the end of the interrupt service routine so that the ST9 is returned to the original status. This occurs for all interrupts and, when operating in the nested mode, up to seven versions of the flag register may be stored.

FLAGR R231 (E7h) System Read/Write Flag Register

Reset value: undefined

7							0
C	Z	S	V	DA	H	UF	DP

b7 = **C:** *Carry Flag.* The carry flag C is affected by the following instructions:

Addition (`add, addw, adc, adcw`),
Subtraction (`sub, subw, sbc, sbcw`),
Compare (`cp, cpw`),
Shift Right Arithmetic (`sra, sraw`),
Rotate (`rrc, rrcw, rlc, rlcw, ror, rol`),
Decimal Adjust (`da`),
Multiply and Divide (`mul, div, divws`).

When set, it generally indicates a carry out of the most significant bit position of the register being used as an accumulator (bit 7 for byte and bit 15 for word operations).

The carry flag can be set by the Set Carry Flag (`scf`) instruction, cleared by the Reset Carry Flag (`rcf`) instruction, and complemented (changed to "0" if "1", and vice versa) by the Complement Carry Flag (`ccf`) instruction.

b6 = **Z:** *Zero Flag.* The Zero flag is affected by the following instructions:

Addition (`add, addw, adc, adcw`),
Subtraction (`sub, subw, sbc, sbcw`),
Compare (`cp, cpw`),
Shift Right Arithmetic (`sra, sraw`),
Rotate (`rrc, rrcw, rlc, rlcw, ror, rol`),
Decimal Adjust (`da`),
Multiply and Divide (`mul, div, divws`),
Logical (`and, andw, or, orw, xor, xorw, cpl`),
Increment and Decrement (`inc, incw, dec, decw`),
Test (`tm, tmw, tcm, tcmw, btset`).

In most cases, the Zero flag is set when the register being used as an accumulator register is zero, following one of the above operations.

b5 = **S:** *Sign Flag.* The Sign flag is affected by the same instructions as the Zero flag.

The Sign flag is set when bit 7 (for byte operation) or bit 15 (for word operation) of the register used as an accumulator is one.

b4 = **V:** *Overflow Flag.* The Overflow flag is affected by the same instructions as the Zero and Sign flags.

When set, the Overflow flag indicates that a two's-complement number, in a result register, is in error, since it has exceeded the largest (or is less than the smallest), number that can be represented in twos-complement notation.

b3 = **DA:** *Decimal Adjust Flag.* The Decimal Adjust flag is used for BCD arithmetic. Since the algorithm for correcting BCD operations is different for addition and subtraction, this flag is used to specify which type of instruction was executed last, so that the subsequent Decimal Adjust (`da`) operation can perform its function correctly.
The Decimal Adjust flag cannot normally be used as a test condition by the programmer.

b2 = **H:** *Half Carry Flag.* The Half Carry flag indicates a carry out of (or a borrow into) bit 3, as the result of adding or subtracting two 8-bit bytes, each representing two BCD digits. The Half Carry flag is used by the Decimal Adjust (`da`) instruction to convert the binary result of a previous addition or subtraction into the correct BCD result.
Like the Decimal Adjust flag, this flag is not normally accessed by the user.

b1 = **UF:** *User Flag.* Bit 1 in the flag register (UF) is available to the user, but it must be set or cleared by an instruction.

b0 = **DP:** *Data/Program Memory Flag.* This bit in the flag register indicates which memory area is addressed. Its value is affected by the Set Data Memory (`sdm`) and Set Program Memory (`spm`) instructions.

If the bit is set, the ST9 addresses the Data Memory Area; when the bit is cleared, the ST9 addresses the Program Memory Area. By reading this bit, the user can verify in which memory area the processor is working. The user writes this bit with the `sdm` or `spm` instructions.

SYSTEM REGISTERS (Continued)

1.1.3.3 Register Pointing Techniques

Two registers, R232 and R233, within the system register group, are available for register pointing. R232 and R233 may be used together as a single pointer for a 16 register working space or separately for two 8 register spaces, in which case R232 becomes Register Pointer 0 (RP0) and R233 becomes Register Pointer 1 (RP1).

The instructions `srp`, `srp0` and `srp1` (the Set Register Pointer instructions) automatically inform the ST9 whether the Register File is to operate with a single 16-register group or two 8-register groups. The `srp0` and `srp1` instructions automatically set the twin 8-register group mode while the `srp` instruction sets the single 16-register group mode. There is no limitation on the order or positions of these chosen register groups other than they must be on 8 or 16 register boundaries.

The addressing of working registers involves use of the Register Pointer value plus an offset value given by the number of the addressed working register.

When addressing a register, the most significant nibble (bits 4-7) gives the group address and the least significant nibble (bits 0-3) gives the register within that group.

REGISTER POINTER 0

RP0 R232 (E8h) System Read/Write Register Pointer 0

Reset Value : undefined

7							0
RG7	RG6	RG5	RG4	RG3	RPS	D1	D0

b7-b3 = **RG7-RG3:** *Register Group number.* These bits contain the number (from 0 to 31) of the group of working registers indicated in the instructions `srp0` or `srp`. When using a 16-register group, a number between 0 and 31 must be used in the `srp` instruction indicating one of the two adjacent 8-register group of working registers used. RG7 is the MSB.

b2 = **RPS:** *Register Pointer Selector.* This bit is set by the instructions `srp0` and `srp1` to indicate that a double register pointing mode is used. Otherwise, the instruction `srp` resets the RPS bit to zero to indicate that a single register pointing mode is used.

b1,b0 = **D1,D0:** These bits are fixed by hardware to zero and are not affected by any writing instruction trying to modify their value.

REGISTER POINTER 1

RP1 R233 (E9h) System Read/Write Register Pointer 1

Reset Value : undefined

7							0
RG7	RG6	RG5	RG4	RG3	RPS	D1	D0

This register is used only with double register pointing mode; otherwise, using single register pointing mode, the RP1R register has to be considered as reserved and not usable as a general purpose register.

b7-b3 = **RG7-RG3:** *Register Group number.* These bits contain the number (from 0 to 31) of the group of 8 working registers indicated in the instructions `srp1`. Bit 7 is the MSB.

b2 = **RPS:** *Register Pointer Selector.* This bit is automatically set by the instructions `srp0` and `srp1` to indicate that a double register pointing mode is used. Otherwise the instruction `srp` reset the RPS bit to zero to indicate that a single register pointing mode is used.

b1,b0 = **D1,D0:** These bits are hardware fixed to zero and are not affected by any writing instruction trying to modify their value.

Note. If working in twin 8-register group mode but only using `srp0` (i.e. only using one 8-register group) the unused register (R233) is to be considered as reserved and not usable as a general purpose register.

The group of registers immediately below the system registers (i.e. group D, R208-R223) can only be accessed via the Register Pointers. To address group D then, it is necessary to set the Register Pointer to group D and then use the addressing procedure for working registers. The programmer is required to remember that the group D should be used as a stacking area. This point is also covered in the Stack Pointers paragraph.

SYSTEM REGISTERS (Continued)

EXAMPLES

Using the Single 16 Register Group

When the system is operating in the single 16-register group mode, the registers are referred to as r0-r15. In this mode, the offset value (i.e. the number of the working register referred to) is supplied in the address (preceded by a small r, e.g. r5) and is added to the Register Pointer 0 value to give the absolute address.

For example, if the Register Pointer contains the value 70h, then working register r7 would have the absolute address, R77h.

In this mode, the single 16-registers group will always start from the lowest even number equal or lower to the number given in the instruction.

Example: `srp #3` is equivalent to `srp #2`.

Using the Twin 8-Register Group

When working in the twin working group mode, the registers pointed by Register Pointer 0 (RP0R), are referred as r0-r7 and those pointed by Register Pointer 1 (RP1R), are referred to as r8-r15, regardless of their absolute addresses. In this mode, when operating with the first 8 working registers (i.e. r0 - r7) the working register number acts as an offset which is added to the value in Register Pointer 0.

So if Register Pointer 0 contains the value 96, then working register 0 has the absolute address 96, working register 5 has the absolute address 101, and so on. The second group of working registers, r8-r15, has the offset values 0 to 7 respectively (i.e. r8 has the offset value 0, r9 has the offset value 1, and so on), this offset value being added to the value in Register Pointer 1.

For example, given that the value in Register Pointer 1 is 32, then working register 12 supplies an offset value of 4 (given by 12 minus 8) to the value in Register Pointer 1 to give an absolute address of 36.

Figure 1-9. Single 16 Register pointing Mode

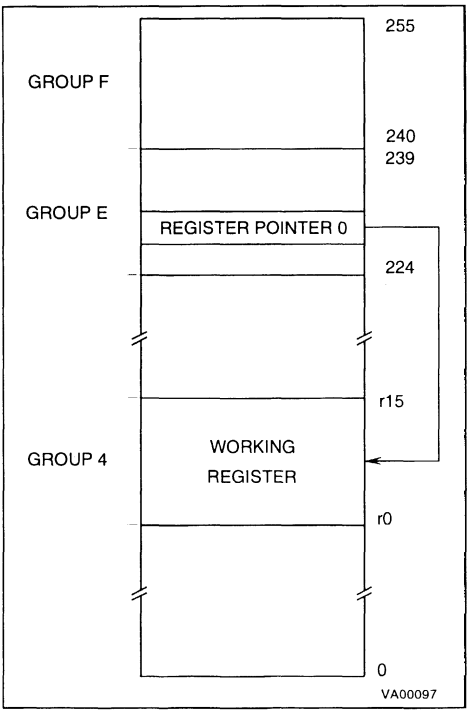
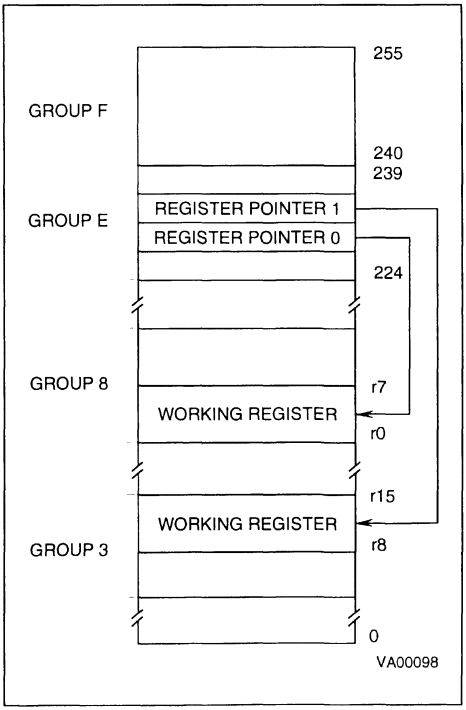


Figure 1-10. Double Register pointing Mode



SYSTEM REGISTERS (Continued)

1.1.3.4 Page Configuration

The pages are available to be used for the storage of control information (such as interrupt vector pointers) relevant to particular peripherals. There are up to 64 pages (each with 16 registers) based on registers R240-R255. These paged registers are addressable via the page pointer register (PPR), which is system register R234.

To address a paged register the page pointer register (R234) must be loaded with the relevant page number using the `spp` instruction (Set Page Pointer) and subsequently any address from the top (F) group (R240-R255) will be referred to that page.

For example if register 23 contains the value 44, the following sequence loads the third register R242 on page 5 with the value 44.

```
spp 5
ld R242, R23
```

PPR R234 (EAh) System Read/Write Page Pointer Register

Reset value : undefined

7							0
PP7	PP6	PP5	PP4	PP3	PP2	D1	D0

b7-b2 = **PP7-PP2:** *Page Pointer.* These bits contain the number (between 0 to 63) of the page chosen by the instruction `ssp` (Set Page Pointer). PP7 is the MSB of the page address. Once the page pointer has been set, there is no need to refresh it unless a different page is required.

b1-b0 = **D1,D0:** These bits are fixed by hardware to zero and are not affected by any writing instruction trying to modify their value.

PAGE 0 contains the control registers of:

- the external interrupt
- the watchdog timer
- the wait logic states
- the serial peripheral interface (SPI)

1.1.3.5 Mode Registers

This register MODER is located in the System Register Group at the address 235.

Using this register it is possible:

- to select either internal or external System and User Stack area,
- to manage the clock frequency
- to enable the Bus request and Wait signals when interfacing external memory.

MODER R235 (EBh) System Read/Write Mode Register

Reset value : 1110 0000

7							0
SSP	USP	DIV2	PRS2	PRS1	PRS0	BRQEN	HIMP

b7 = **SSP:** *System Stack Pointer.* This bit selects internal (in the Register File) or external (in the external Data Memory) System Stack area, logical "1" for internal, and logical "0" for external. After Reset the value of this bit is "1".

b6 = **USP:** *User Stack Pointer.* Same as bit 7 for the User Stack Pointer;

b5 = **DIV2:** *OSCIN Clock Divided by 2.* This bit controls the divide by 2 circuit which operates on the OSCIN Clock. A logical "1" value means that the OSCIN clock is internally divided by 2, and a logical "0" value means that no division of the OSCIN Clock occurs.

b4-b2 = **PRS2-PRS0:** *ST9 CPUCLK Prescaler.* These bits load the prescaling module of the internal clock (INTCLK). The prescaling value selects the frequency of the ST9 clock, which can be divided by 1 to 8. See Clock chapter for more information.

b1 = **BRQEN:** *Bus Request Enable.* This bit must be held to "0".

b0 = **HIMP:** *High Impedance Enable.* This bit must be held to "0".

SYSTEM REGISTERS (Continued)

1.1.3.6 Stack Pointers

There are two separate, double register stack pointers available (named System Stack Pointer and User Stack Pointer), both of which can address registers or memory.

The stack pointers point to the bottom of the stacks which are filled using the `push` commands and emptied using the `pop` commands. The stack pointer is automatically pre-decremented when data is "pushed in" and post-incremented when data is "popped out".

For example, the register address space is selected for a stack and the corresponding stack pointer register contains 220. When a byte of data is "pushed" into the stack, the stack pointer register is decremented to 219, then the data byte is "loaded" into register 219. Conversely, if a stack pointer register contains 189 and a byte of data is "popped" out, the byte of data is then extracted from the stack and then the stack pointer register is incremented to 190.

The `push` and `pop` commands used to manage the system stack area are made applicable to the user stack by adding the suffix U, while to use a stack instruction for a word a W is added.

For example `push` inserts data into the system stack, but an added U indicates the user stack and W means a word, so the instruction `pushuw` loads a word into the bottom of the user stack.

If the User Stack Pointer register contains 223 (working in register space) the instruction `pushuw` will decrement User Stack Pointer register to 222 and then load a word into register R222 and R221.

When bytes (or words) are "popped out" the values in those registers are left unchanged until fresh data is loaded into those locations. Thus when data is "popped" out from a stack area, the stack content remains unchanged.

Note. Stacks must not be located in the pages or the system register area.

The System Stack area and The System Stack Pointer

The System Stack area is used for the storage of temporarily suspended system and/or control registers, i.e. the Flag register and the Program counter, while interrupts are being serviced. For subroutine execution only the Program Counter needs to be saved in the System stack area.

There are two situations when this occurs automatically, one being when an interrupt occurs and the other when the instruction call subroutine is used. When the system stack area is in the Register File, the stack pointer, which points to the bottom of the stack, only needs one byte for addressing, in which case the System Stack Pointer Low Register (R239) is sufficient for addressing purposes. As a result the System Stack Pointer High Register (R238) becomes redundant BUT must be considered as reserved (please refer also to "spurious" memory access section). Clearly when the stack is external a full word address is necessary and so both registers are used to point, the even register providing the MSB and the odd register providing the LSB.

The User Stack area and User Stack Pointer

The User Stack area is completely free from all interference from automatic operations and so it provides a totally user controlled stacking area, that area being in any part of the memory which is of a RAM nature, or the first 14 groups of the general Register File i.e. not in the System register or Paged group.

The User Stack Pointer consists of two registers, R236 and R237, which are both used for addressing an external stack, while, when stacking in the Register File, the User Stack Pointer High Register, R236, becomes redundant but must be considered as reserved.

SYSTEM REGISTERS (Continued)

Stack location

Care is necessary when managing stacks as there is no limit to stack sizes apart from the bottom of any address space in which the stack is placed. Consequently programmers are advised to use a stack pointer value as high as possible, particularly when using the Register File as a stacking area. This will also benefit programmers who may locate the stacks in group D using, for example the instruction `ld R237, #223` which loads the value 223 into the User Stack Pointer Low Register. The Programmer will not need to remember to set the Register Pointer to 208 to gain access to registers in the D-group, a problem outlined in Register Pointing Techniques paragraph.

Stacks may be located anywhere in the first 14 groups of the Register File (internal stacks) or the data memory (external stacks). It is not necessary to set the data memory using the instruction `sdm` as external stack instructions automatically use the data memory.

Figure 1-11. System and/or User Stack in Register Stack Mode

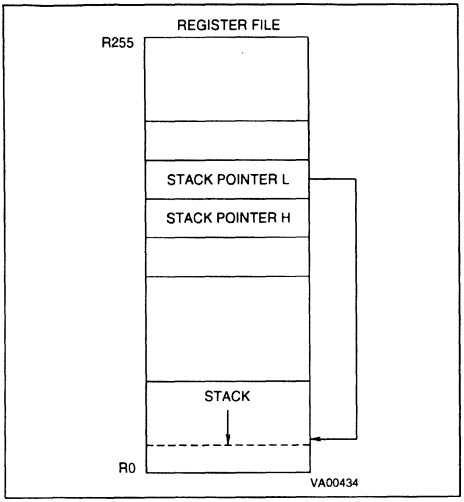
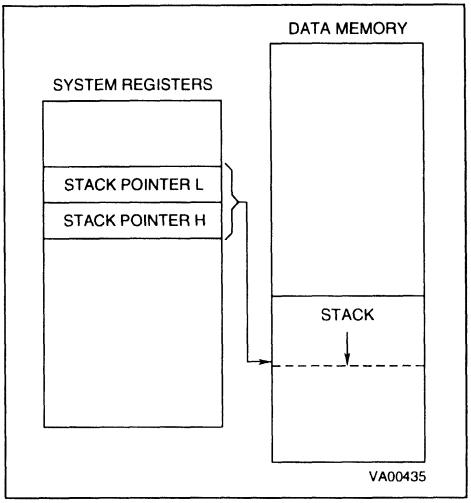


Figure 1-12. System and/or User Stack in Memory Stack Mode



USP R236 (ECh) System Read/Write
User Stack Pointer High Byte

Reset value: undefined

7							0
USP15	USP14	USP13	USP12	USP11	USP10	USP9	USP8

USP R237 (EDh) System Read/Write
User Stack Pointer Low Byte

Reset value: undefined

7							0
USP7	USP6	USP5	USP4	USP3	USP2	USP1	USP0

SSP R238 (EEh) System Read/Write
System Stack Pointer High Byte

Reset value: undefined

7							0
SSP15	SSP14	SSP13	SSP12	SSP11	SSP10	SSP9	SSP8

SSP R239 (EFh) System Read/Write
System Stack Pointer Low Byte

Reset value: undefined

7							0
SSP7	SSP6	SSP5	SSP4	SSP3	SSP2	SSP1	SSP0

1.2 MEMORY

1.2.1 INTRODUCTION

The memory of the ST9291 is functionally divided into two areas, the Register File and Memory. The Memory may optionally be divided into two spaces, Program Memory for Program code and Data Memory for Data.

The memory spaces are selected by the execution of the SDM and SPM instructions (Set Data Memory and Set Program Memory, respectively). There is no need to use either of these instructions again until the memory area required is to be changed.

1.2.1.1 Program Space

The Program memory space of the ST9291 consists of 32K bytes of on-chip ROM (addressed from 0 to 07FFFh) and 640 bytes of on-chip RAM (addressed from FD80h to FFFFh); refer to the memory map tables and drawing on the following page for the memory mapping for other ROM sizes. The first 256 memory locations from address 0 to 00FFh (hexadecimal) hold the Reset Vector, the Top-Level (Pseudo Non-Maskable) interrupt, the Divide by Zero Trap vector and, optionally, the interrupt vector table for use with the on-chip peripherals and the external interrupt sources. Each vector is contained in two consecutive byte locations, the high order address held in the lower (even) byte, the low order address held in the upper (odd) byte, forming the address which is loaded into the Program Counter when selected by the interrupt vector provided by the interrupt source. This should point to the relevant Interrupt Service routine provided by the User for immediate response to the interrupt.

1.2.1.2 Data Space

The ST9291 addresses the 640 bytes of on-chip RAM memory from addresses FD80h to FFFFh in both Program and Data Space. On-chip general purpose Registers may be used as additional RAM memory for minimum chip count systems.

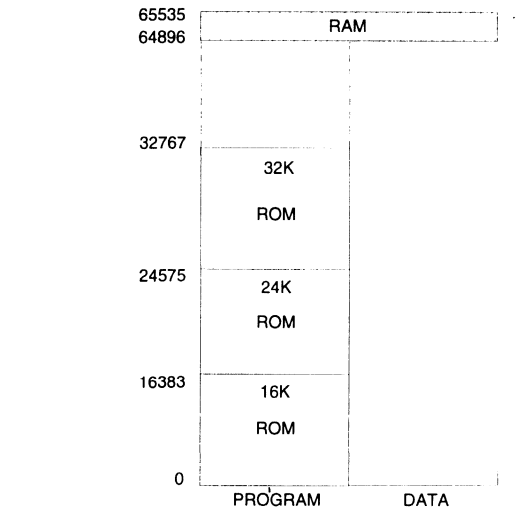
The Data Space is selected by the execution of the SDM instruction. All subsequent memory references will access the Data Space. When a separate Data Space is not required, data may stored in RAM or ROM memory within the Program Space.

MEMORY (Continued)

Table 1-4. ROM and RAM Address Configuration

Device Suffix	ROM Size (Bytes)	ROM Addresses		RAM Size (Bytes)	RAM Addresses	
J6/N6	32K	0 - 32767	dec	640	64896 - 65535	dec
		0000 - 7FFF	hex		FD80 - FFFF	hex
J5/N5	24K	0 - 24575	dec	640	64896-65535	dec
		0000 - 5FFF	hex		FD80 - FFFF	hex
J4/N4	24K	0 - 24575	dec	384	65152-65535	dec
		0000 - 5FFF	hex		FE80 - FFFF	hex
J3/N3	16K	0 - 16383	dec	640	64896-65535	dec
		0000 - 3FFF	hex		FD80 - FFFF	hex
J2/N2	16K	0 - 16383	dec	384	65152-65535	dec
		0000 - 3FFF	hex		FE80 - FFFF	hex

Figure 1-13. ST9291 Memory Map



GENERAL INDEX

Pages

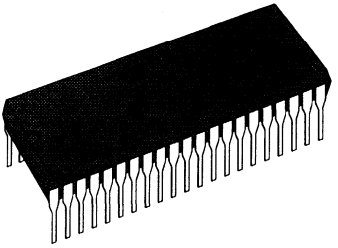
SGS-THOMSON MICROELECTRONICS

ST9293

48K ROM HCMOS MCUs WITH ON SCREEN DISPLAY AND A/D CONVERTER

PRELIMINARY DATA

- Register oriented 8/16 bit CORE with RUN, WFI and HALT modes
- Minimum instruction cycle time: 500ns (12MHz internal)
- 16 to 48K bytes of ROM, 256 to 768 bytes of RAM, 224 general purpose registers available as RAM, accumulators or index registers (Register File)
- 42-lead Shrink DIP package
- Interrupt handler and Serial Peripheral Interface as standard features
- 31 fully programmable I/O pins
- 34 character x15 rows software programmable On Screen Display module with colour, italic, underline, flash, transparent and fringe attribute options
- 16 bit Timer with 8 bit Prescaler, able to be used as a Watchdog Timer
- 16-bit programmable Slice Timer with 8-bit prescaler
- 4 channel Analog to Digital Converter, with integral sample and hold, fast 5.75µs conversion time, 6-bit guaranteed resolution
- Rich Instruction Set and 14 Addressing modes
- Division-by-Zero trap generation
- Versatile Development tools, including assembler, linker, C-compiler, archiver, graphic oriented debugger and hardware emulators
- Real Time Operating System
- Windowed EPROM parts available for prototyping and pre-production development phases



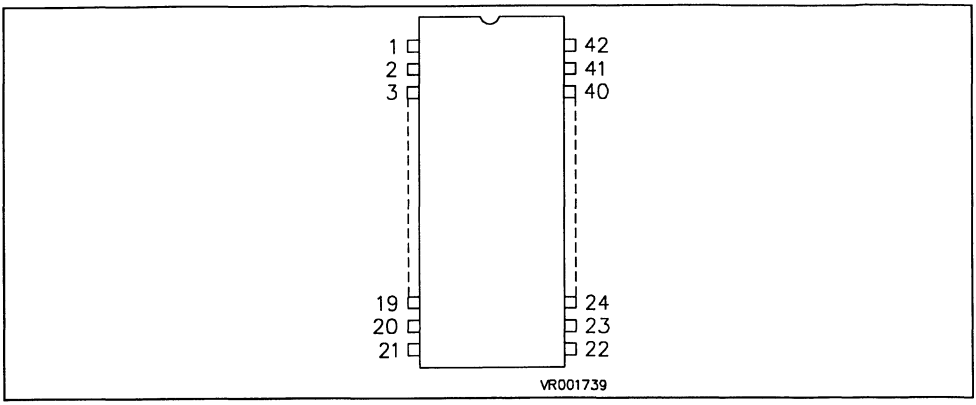
PSDIP42

(Ordering Information at the end of the Datasheet)

DEVICE SUMMARY

Device	ROM	RAM	PACKAGE
ST9293J7	48K	768	PSDIP42
ST9293J5	32K	640	PSDIP42
ST9293J3	24K	512	PSDIP42
ST9293J1	16K	256	PSDIP42

Figure 1. 42 Pin Shrink DIP Pinout



ST9293J Pin Description

Pin	Pin name
1	P2.4/WDOUT/EXTRG (AD)
2	P2.5/SDO
3	P2.6/SCK/INT2
4	P2.7/SDI/SDO
5	P0.7
6	P0.6
7	P0.5
8	P0.4
9	P0.3
10	P0.2
11	P0.1
12	P0.0
13	V_{DD}
14	P5.0
15	P5.1
16	P5.2
17	P5.3/FB
18	P5.4/B
19	P5.5/G
20	P5.6/R
21	V_{SS}

Pin	Pin name
42	P2.3/INT3
41	P2.2/INT0
40	P2.1/INT7
39	P2.0/INT6
38	P3.7/NMI
37	P3.6/AIN4/WDIN
36	P3.5/INT4
35	P3.4-OD/SLIN
34	P3.6-OD/SLOUT
33	$\overline{\text{RESET}}$
32	OSCIN
31	V_{SS2}
30	OSCOUT
29	P4.7/AIN7
28	P4.6/AIN6
27	P4.5/AIN5
26	VSYNC
25	HSYNC
24	AV_{DD}
23	PLLR
22	PLLF

1.1GENERAL DESCRIPTION

The ST9293 is a ROM member of the ST9 family of microcontrollers, completely developed and produced by SGS-THOMSON Microelectronics using a proprietary n-well HCMOS process.

The ROM parts are fully compatible with their EPROM versions, which may be used for the prototyping and pre-production phases of development, and can be configured as standalone microcontrollers with 48K/32K/24K/16 bytes of on-chip ROM.

The nucleus of the ST9293 is the advanced Core which includes the Central Processing Unit (CPU), the Register File, a 16-bit Timer/Watchdog with 8-bit Prescaler, a Serial Peripheral Interface supporting S-bus, I^2C-bus and IM-bus Interface, plus two 8-bit I/O ports. The Core has independent memory and register buses allowing a high degree of pipelining to add to the efficiency of the code execution speed of the extensive instruction set.The powerful I/O capabilities demanded by microcontroller applications are fulfilled by the ST9293 with up to 31/41 I/O lines dedicated to digital Input/Output.

These lines are grouped into up to six I/O Ports and can be configured on a bit basis under software control to provide timing, status signals, timer inputs and outputs, analog inputs, external interrupts and serial or parallel I/O.

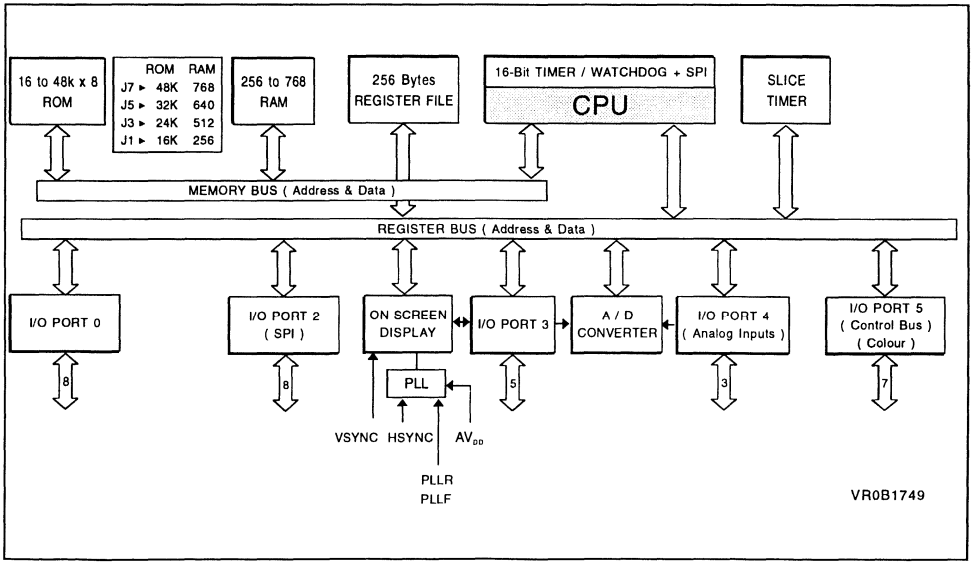
Three basic memory spaces are available to support this wide range of configurations: Program Memory, Data Memory and the Register File, which includes the control and status registers of the on-chip peripherals.

The 16-bit Slice Timer with an 8-bit Prescaler and 6 operating modes allows simple use for waveformgeneration and measurement, PWM functions and many other system timing functions.

The human interface is provided by the On Screen Display module, this can produce up to 8 lines of of up to 34 characters from a ROM defined 128 character set. The 9x13 character can be modified by 4 different pixel sizes, with character rounding, and formed into words with colour and format attributes.

In addition there is a 4 channel Analog to Digital Converter with integral sample and hold, fast 5.5µs conversion time and 6-bit guaranteed resolution.

Figure 1-2. ST9293 Block Diagram



1.2 PIN DESCRIPTION

VSYNC. *Vertical Synch.* Vertical video synchronisation input to OSD. Positive or negative polarity.

HSYNC. *Horizontal Synch.* Horizontal video synchronisation input to OSD. Positive or negative polarity.

PLLF. *PLL Filter input.* Filter input for the OSD for PLL feed-back.

PLLR. *PLL Resistor connection pin.* For resistor connection to select the PLL gain adjust.

$\overline{\text{RESET}}$. *Reset (input, active low).* The ST9 is initialised by the Reset signal. With the deactivation of $\overline{\text{RESET}}$, program execution begins from the Program memory location pointed to by the vector contained in program memory locations 00h and 01h.

OSCIN, OSCOUT. *Oscillator (input and output).* These pins connect a parallel-resonant crystal (24MHz maximum), or an external source to the on-chip clock oscillator and buffer. OSCIN is the input of the oscillator inverter and internal clock generator; OSCOUT is the output of the oscillator inverter.

AV_{DD}. Analog V_{DD} of PLL. This pin must be tied to V_{DD} externally to the ST9293.

V_{DD}. Main Power Supply Voltage (5V±10%)

V_{SS}, V_{SS2}. Digital Circuit Ground, these pins must be connected together externally to the ST9293.

P0.0-P0.7, P2.0-P2.7, P3.3-P3.7, P4.5-P4.7, P5.0-P5.6 *I/O Port Lines (Input/Output, TTL or CMOS compatible).* 31 lines grouped into I/O ports, bit programmable under program control as general purpose I/O or as Alternate functions (see next section).
P3.3 and P3.4 are true 12V open drain outputs when set in output mode.

1.2.1 I/O Port Alternate Functions.

Each pin of the I/O ports of the ST9293 may assume software programmable Alternative Functions as shown in the Pin Configuration Drawings. Table 1 shows the Functions allocated to each I/O Port pin.

Table 1-1. ST9293 I/O Port Alternate Function Summary

I/O PORT	Name	Function	Alternate Function	Pin Assignment
Port.bit				9293J
P0.0		I/O		12
P0.1		I/O		11
P0.2		I/O		10
P0.3		I/O		9
P0.4		I/O		8
P0.5		I/O		7
P0.6		I/O		6
P0.7		I/O		5
P2.0	INT6	I	External Interrupt 6	39
P2.1	INT7	I	External Interrupt 7	40
P2.2	INT0	I	External Interrupt 0	41

PIN DESCRIPTION (Continued)

Table 1-2. ST9293 I/O Port Alternate Function Summary

I/O PORT	Name	Function	Alternate Function	Pin Assignment
Port.bit				9293J
P2.3	INT3	I	External Interrupt 3	42
P2.4	WDOUT	O	T/WD Output	1
P2.4	EXTRG	I	External A/D Trigger	1
P2.5	SDO	O	MSPI Serial Data Output	2
P2.6	INT2	I	External Interrupt 2	3
P2.6	SCK	O	SPI Serial Clock	3
P2.7	SDO	O	SPI Serial Data Output	4
P2.7	SDI	I	SPI Serial Data Input	4
P3.3		O	(12V Open Drain Output)	34
P3.3	SLOUT	O	Slice Timer Output	34
P3.4		O	(12V Open Drain Output)	35
P3.4	SLIN	I	Slice Timer Input	35
P3.5	INT4	I	Schmitt Triggered Input Only	36
P3.6	WDIN	I	T/WD Input	37
P3.6	AIN4	I	A/D Analog Input 4	37
P3.7	NMI	I	Non-Maskable Interrupt	38
P4.5	AIN5	I	A/D Analog Input 5	27
P4.6	AIN6	I	A/D Analog Input 6	28
P4.7	AIN7	I	A/D Analog Input 7	29
P5.0		I/O		14
P5.1		I/O		15
P5.2		I/O		16
P5.3	FB	O	Fast Blanking OSD output	17
P5.4	B	O	Blue Video Colour OSD output	18
P5.5	G	O	Green Video Colour OSD output	19
P5.6	R	O	Red Video Colour OSD output	20

NOTES

2 CORE ARCHITECTURE

2.1 CORE ARCHITECTURE

The Core or Central Processing Unit (CPU) of the ST9 includes the 8 bit Arithmetic Logic Unit and the 16 bit Program Counter, System and User Stack Pointers. The microcoded Instruction Set is highly optimised for both byte (8 bit) and word (16 bit) data, BCD and Boolean data types, with 14 addressing modes.
Three independent buses are controlled by the Core, a 16 bit Memory bus, an 8 bit Register addressing bus and a 6 bit Interrupt/DMA bus connected to the interrupt and DMA controllers in the on-chip peripherals and the Core. This multiple bus architecture allows a high degree of pipelining and parallel operation, giving the ST9 its efficiency in both numerical calculations and communication with the on-chip peripherals.

2.2 ADDRESS SPACES

The ST9 has three separate address spaces:

- Register File: 240 8-bit registers plus up to 64 pages of 16 bytes each, located in the on-chip peripherals.
- Data memory with up to 64K (65536) bytes
- Program memory with up to 64K (65536) bytes

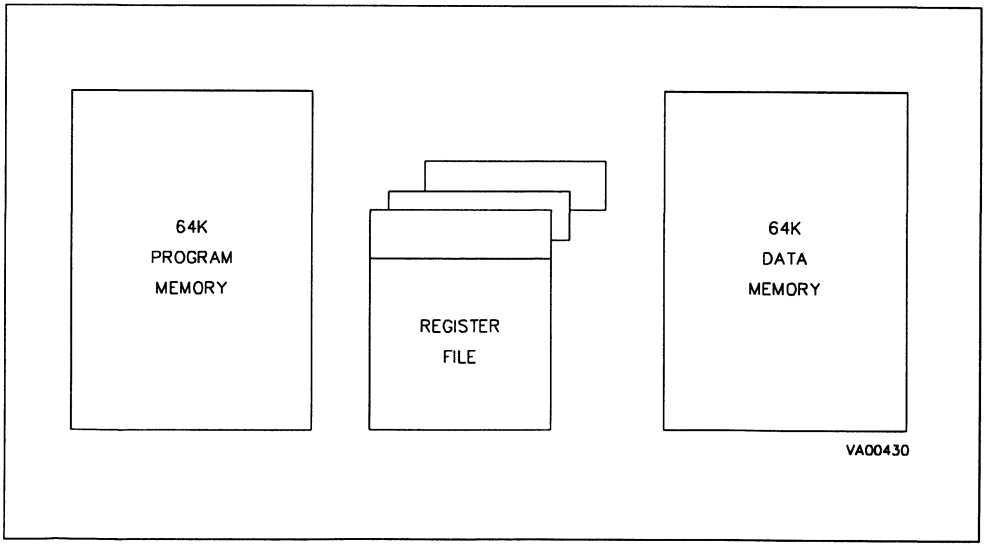
The Data and Program memory spaces will be addressed in further detail in the next section.

2.2.1 Register File

The Register File consists of:

- 224 general purpose registers R0 to R223
- 16 system registers in the System Group (R224 to R239).
- I/O pages depending on the configuration of the ST9, each containing up to 16 registers, with paging facilities based on the top group (R240 to R255).

Figure 2-1. Address Spaces



ADDRESS SPACES (Continued)

Figure 2-2. Register Grouping

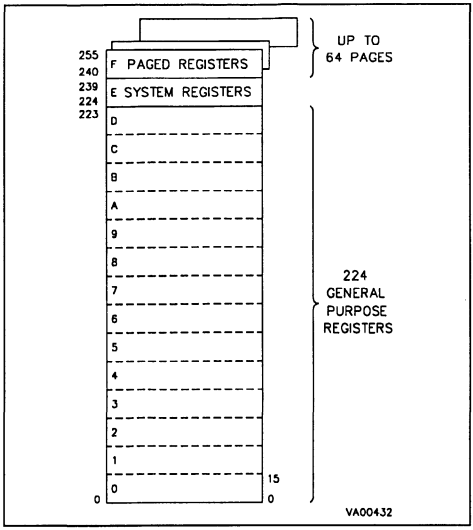


Figure 2-3. Page Pointer Configuration

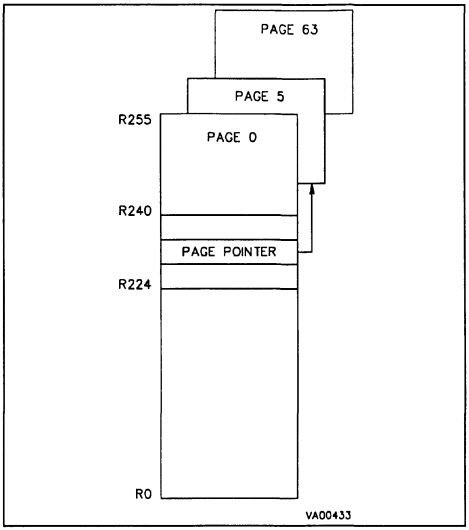
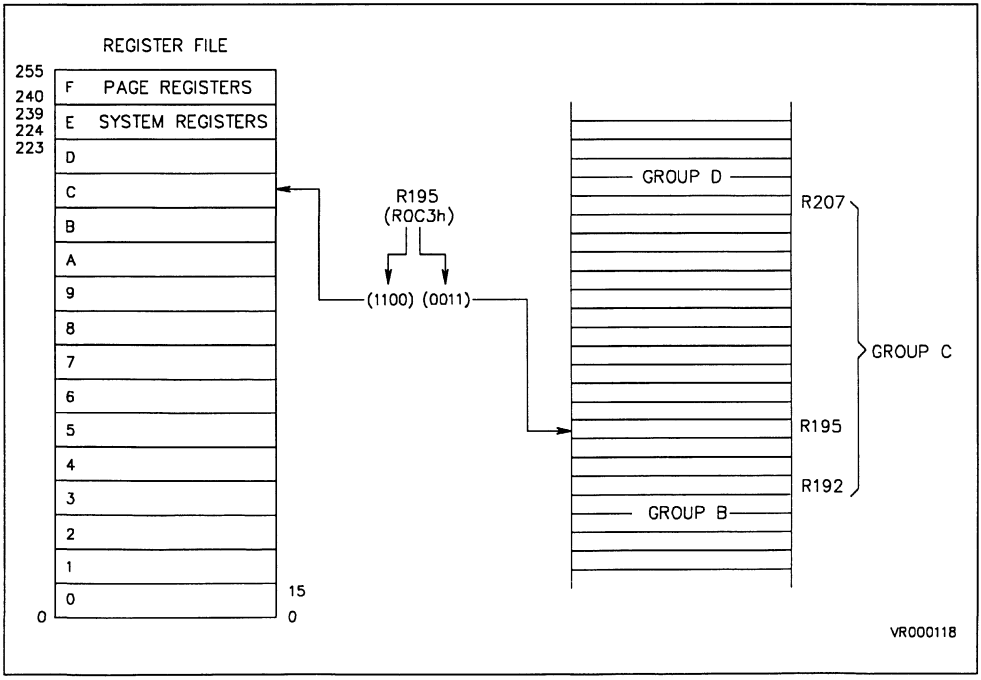


Figure 2-4. Addressing the Register File



ADDRESS SPACES (Continued)

2.2.2 Addressing Registers

All registers in the Register File and pages can be specified by using a decimal, hex or binary address, e.g. R231, RE7h or R11100111b is the same register.
The registers can be referred to by their hexadecimal group address, so that registers R0-R15 form group 0, R160-R175 form group A and so on.

Working Register Addresses

The 8-bit register address is formed by 2 nibbles, for example, for register R195 or RC3h or R11000011, 1100 specifies the 13th group (i.e. group C) and 0011 specifies the 3rd register in that group.

Working registers are addressed by supplying the least significant nibble in the instruction and adding it to the most significant nibble found in the Register Pointer (R233). Working register addressing is shown in Figures 2-4.

System Registers

The 16 system registers at addresses R224 to R239 form Group E.
The system registers are addressable using any of the 4 register addressing modes and the most significant nibble will, in all cases, be 14 (0Eh).

Paged Registers

There are a maximum of 64 pages each containing 16 registers. These are addressed using the register addressing modes with the addition of the Page Pointer register, R234. This register selects the page to be addressed in group F and once set, does not need to be changed if two or more registers on the same page are to be addressed in succession.

Therefore if the Page Pointer, R234, is set to 5, the instructions

```
spp 5
ld R242, r4
```

will load the contents of working register r4 into the third register (R242) of page 5.

These paged registers hold data and control registers related to the on-chip peripherals, and thus the configuration depends upon the peripheral organisation of each ST9 family member. i.e. pages only exist if the peripheral exists.

Available pages are shown in Table 2-2.

2.2.3 Input/Output Ports

The Input/Output ports are located in two areas. The port registers for Ports 0-5 are located at the bottom of the System register group in locations R224 to R229.

Each Port has three associated Control registers, which determine the individual pin modes (I/O, Open-Drain etc). These registers are located in pages 2 and 3.

Table 2-1. Register File Organization

Hex. Address	Decimal Address	Function	Register File Group
F0-FF	240-255	Paged Registers	Group F
E0-EF	224-239	System Registers	Group E
D0-DF	208-223	General Purpose Registers	Group D
C0-CF	192-207		Group C
B0-BF	176-191		Group B
A0-AF	160-175		Group A
90-9F	144-159		Group 9
80-8F	128-143		Group 8
70-7F	112-127		Group 7
60-6F	96-111		Group 6
50-5F	80-95		Group 5
40-4F	64-79		Group 4
30-3F	48-63		Group 3
20-2F	32-47		Group 2
10-1F	16-31		Group 1
00-0F	00-15		Group 0

ADDRESS SPACES (Continued)

Table 2-2. Group F Peripheral Organization

Applicable for ST9293

DEC	DEC HEX	00 00	02 02	03 03	0B 11	28 40	29 41	2A 42	3E 62
R255	RFF	RES.	RES.	RES.	RES.	OSD CHAR 1 TO 16	OSD CHAR 17 TO 32	RES.	RES.
R254	RFE	SPI	PORT 3						
R253	RFD								
R252	RFC	WCR							
R251	RFB	T/WD	RES.					OSD	
R250	RFA		PORT 2						
R249	RF9								
R248	RF8								
R247	RF7	EXT INT	RES.						
R246	RF6		PORT1	PORT 5					
R245	RF5								
R244	RF4								
R243	RF3		RES.	RES.	SLICE TIMER			OSD CHAR 33 TO 36	A/D CONV
R242	RF2		PORT 0	PORT 4					
R241	RF1	RES.							
R240	RF0								

2.3 SYSTEM REGISTERS

Following is the description of System Registers. For PORT0 to PORT5 Registers, please refer to I/O Port Chapter.

Figure 2-5. System Registers

R239 (EFh)	SYS. STACK POINTER LOW
R238 (EEh)	SYS. STACK POINTER HIGH
R237 (EDh)	USER STACK POINTER LOW
R236 (ECh)	USER STACK POINTER HIGH
R235 (EBh)	MODE REGISTER
R234 (EAh)	PAGE POINTER
R233 (E9h)	REGISTER POINTER 1
R232 (E8h)	REGISTER POINTER 0
R231 (E7h)	FLAGS
R230 (E6h)	CENTRAL INT. CNTL REG
R229 (E5h)	PORT5
R228 (E4h)	PORT4
R227 (E3h)	RESERVED
R226 (E2h)	PORT2
R225 (E1h)	PORT1
R224 (E0h)	PORT0

2.3.1 Central Interrupt Control Register

This Register CICR is located in the system Register Group at the address R230 (E6h). Please refer to "INTERRUPT" and "DMA" chapters in order to get the background of the ST9 interrupt philosophy.

CICR R230 (E6h) System Read/Write
Central Interrupt Control Register

Reset Value : 1000 0111

7							0
GCEN	TLIP	TLI	IEN	IAM	CPL2	CPL1	CPL0

b7 = **GCEN:** *Global Counter Enable.* This bit is the Global Counter Enable of the Multifunction Timers. The GCEN bit is ANDed with the CE (Counter Enable) bit of the Timer Control Register (explained in the Timer chapter) in order to enable the Timers when both bits are set. This bit is set after the Reset cycle.

b6 = **TLIP:** *Top Level Interrupt Pending.* This bit is automatically set when a Top Level Interrupt Request is recognized. This bit can also be set by Software in order to simulate a Top Level Interrupt Request.

b5 = **TLI:** *Top Level Interrrupt bit.* When this bit is set, a Top Level interrupt request is acknowledged depending on the IEN bit and the TLNM bit (in Nested Interrupt Control Register). If the TLM bit is reset the top level interrupt acknowledgement depends on the TLNM alone.

b4 = **IEN:** *Enable Interrupt.* This bit, (when set), allows interrupts to be accepted. When reset no interrupts other than the NMI can be acknowledged. It is cleared by interrupt acknowledgement for concurrent mode and set by interrupt return (`iret`). It can be managed by hardware and software (`ei` and `di` instruction).

b3 = **IAM:** *Interrupt Arbitration Mode.* This bit covers the selection of the two arbitration modes, the Concurrent Mode being indicated by the value "0" and the Fully Automatic Nested Mode by the value "1". This bit is under software control.

b2-b0 = **CPL2-CPL0:** *Current Priority Level.* These three bits record the priority level of the interrupt presently under service (i.e. the Current Priority Level, CPL). For these priority levels 000 is the highest priority and 111 is the lowest priority. The CPL bits can be set by hardware or software and give the reference by which following interrupts are either left pending or able to interrupt the current interrupt. When the present interrupt is replaced by one of a greater priority, the current priority value is automatically stored until required.

SYSTEM REGISTERS (Continued)

2.3.2 Flag Register

The Flag Register contains 8 flags indicating the status of the ST9. During an interrupt the flag register is automatically stored in the system stack area and recalled at the end of the interrupt service routine so that the ST9 is returned to the original status. This occurs for all interrupts and, when operating in the nested mode, up to seven versions of the flag register may be stored.

FLAGR R231 (E7h) System Read/Write Flag Register

Reset value: undefined

7							0
C	Z	S	V	DA	H	UF	DP

b7 = **C:** *Carry Flag.* The carry flag C is affected by the following instructions:

Addition (`add, addw, adc, adcw`),
Subtraction (`sub, subw, sbc, sbcw`),
Compare (`cp, cpw`),
Shift Right Arithmetic (`sra, sraw`),
Rotate (`rrc, rrcw, rlc, rlcw, ror, rol`),
Decimal Adjust (`da`),
Multiply and Divide (`mul, div, divws`).

When set, it generally indicates a carry out of the most significant bit position of the register being used as an accumulator (bit 7 for byte and bit 15 for word operations).

The carry flag can be set by the Set Carry Flag (`scf`) instruction, cleared by the Reset Carry Flag (`rcf`) instruction, and complemented (changed to "0" if "1", and vice versa) by the Complement Carry Flag (`ccf`) instruction.

b6 = **Z:** *Zero Flag.* The Zero flag is affected by the following instructions:

Addition (`add, addw, adc, adcw`),
Subtraction (`sub, subw, sbc, sbcw`),
Compare (`cp, cpw`),
Shift Right Arithmetic (`sra, sraw`),
Rotate (`rrc, rrcw, rlc, rlcw, ror, rol`),
Decimal Adjust (`da`),
Multiply and Divide (`mul, div, divws`),
Logical (`and, andw, or, orw, xor, xorw, cpl`),
Increment and Decrement (`inc, incw, dec, decw`),
Test (`tm, tmw, tcm, tcmw, btset`).

In most cases, the Zero flag is set when the register being used as an accumulator register is zero, following one of the above operations.

b5 = **S:** *Sign Flag.* The Sign flag is affected by the same instructions as the Zero flag.

The Sign flag is set when bit 7 (for byte operation) or bit 15 (for word operation) of the register used as an accumulator is one.

b4 = **V:** *Overflow Flag.* The Overflow flag is affected by the same instructions as the Zero and Sign flags.

When set, the Overflow flag indicates that a two's-complement number, in a result register, is in error, since it has exceeded the largest (or is less than the smallest), number that can be represented in twos-complement notation.

b3 = **DA:** *Decimal Adjust Flag.* The Decimal Adjust flag is used for BCD arithmetic. Since the algorithm for correcting BCD operations is different for addition and subtraction, this flag is used to specify which type of instruction was executed last, so that the subsequent Decimal Adjust (`da`) operation can perform its function correctly.
The Decimal Adjust flag cannot normally be used as a test condition by the programmer.

b2 = **H:** *Half Carry Flag.* The Half Carry flag indicates a carry out of (or a borrow into) bit 3, as the result of adding or subtracting two 8-bit bytes, each representing two BCD digits. The Half Carry flag is used by the Decimal Adjust (`da`) instruction to convert the binary result of a previous addition or subtraction into the correct BCD result.
Like the Decimal Adjust flag, this flag is not normally accessed by the user.

b1 = **UF:** *User Flag.* Bit 1 in the flag register (UF) is available to the user, but it must be set or cleared by an instruction.

b0 = **DP:** *Data/Program Memory Flag.* This bit in the flag register indicates which memory area is addressed. Its value is affected by the Set Data Memory (`sdm`) and Set Program Memory (`spm`) instructions.

If the bit is set, the ST9 addresses the Data Memory Area; when the bit is cleared, the ST9 addresses the Program Memory Area. By reading this bit, the user can verify in which memory area the processor is working. The user writes this bit with the `sdm` or `spm` instructions.

SYSTEM REGISTERS (Continued)

2.3.3 Register Pointing Techniques

Two registers, R232 and R233, within the system register group, are available for register pointing. R232 and R233 may be used together as a single pointer for a 16 register working space or separately for two 8 register spaces, in which case R232 becomes Register Pointer 0 (RP0) and R233 becomes Register Pointer 1 (RP1).

The instructions `srp`, `srp0` and `srp1` (the Set Register Pointer instructions) automatically inform the ST9 whether the Register File is to operate with a single 16-register group or two 8-register groups. The `srp0` and `srp1` instructions automatically set the twin 8-register group mode while the `srp` instruction sets the single 16-register group mode. There is no limitation on the order or positions of these chosen register groups other than they must be on 8 or 16 register boundaries.

The addressing of working registers involves use of the Register Pointer value plus an offset value given by the number of the addressed working register.

When addressing a register, the most significant nibble (bits 4-7) gives the group address and the least significant nibble (bits 0-3) gives the register within that group.

REGISTER POINTER 0

RP0 R232 (E8h) System Read/Write Register Pointer 0

Reset Value : undefined

7							0
RG7	RG6	RG5	RG4	RG3	RPS	D1	D0

b7-b3 = **RG7-RG3:** *Register Group number.* These bits contain the number (from 0 to 31) of the group of working registers indicated in the instructions `srp0` or `srp`. When using a 16-register group, a number between 0 and 31 must be used in the `srp` instruction indicating one of the two adjacent 8-register group of working registers used. RG7 is the MSB.

b2 = **RPS:** *Register Pointer Selector.* This bit is set by the instructions `srp0` and `srp1` to indicate that a double register pointing mode is used. Otherwise, the instruction `srp` resets the RPS bit to zero to indicate that a single register pointing mode is used.

b1,b0 = **D1,D0:** These bits are fixed by hardware to zero and are not affected by any writing instruction trying to modify their value.

REGISTER POINTER 1

RP1 R233 (E9h) System Read/Write Register Pointer 1

Reset Value : undefined

7							0
RG7	RG6	RG5	RG4	RG3	RPS	D1	D0

This register is used only with double register pointing mode; otherwise, using single register pointing mode, the RP1R register has to be considered as reserved and not usable as a general purpose register.

b7-b3 = **RG7-RG3:** *Register Group number.* These bits contain the number (from 0 to 31) of the group of 8 working registers indicated in the instructions `srp1`. Bit 7 is the MSB.

b2 = **RPS:** *Register Pointer Selector.* This bit is automatically set by the instructions `srp0` and `srp1` to indicate that a double register pointing mode is used. Otherwise the instruction `srp` reset the RPS bit to zero to indicate that a single register pointing mode is used.

b1,b0 = **D1,D0:** These bits are hardware fixed to zero and are not affected by any writing instruction trying to modify their value.

Note. If working in twin 8-register group mode but only using `srp0` (i.e. only using one 8-register group) the unused register (R233) is to be considered as reserved and not usable as a general purpose register.

The group of registers immediately below the system registers (i.e. group D, R208-R223) can only be accessed via the Register Pointers. To address group D then, it is necessary to set the Register Pointer to group D and then use the addressing procedure for working registers. The programmer is required to remember that the group D should be used as a stacking area. This point is also covered in the Stack Pointers paragraph.

SYSTEM REGISTERS (Continued)

EXAMPLES

Using the Single 16 Register Group

When the system is operating in the single 16-register group mode, the registers are referred to as r0-r15. In this mode, the offset value (i.e. the number of the working register referred to) is supplied in the address (preceded by a small r, e.g. r5) and is added to the Register Pointer 0 value to give the absolute address.

For example, if the Register Pointer contains the value 70h, then working register r7 would have the absolute address, R77h.

In this mode, the single 16-registers group will always start from the lowest even number equal or lower to the number given in the instruction.

Example: `srp #3` is equivalent to `srp #2`.

Using the Twin 8-Register Group

When working in the twin working group mode, the registers pointed by Register Pointer 0 (RP0R), are referred as r0-r7 and those pointed by Register Pointer 1 (RP1R), are referred to as r8-r15, regardless of their absolute addresses. In this mode, when operating with the first 8 working registers (i.e. r0 - r7) the working register number acts as an offset which is added to the value in Register Pointer 0.

So if Register Pointer 0 contains the value 96, then working register 0 has the absolute address 96, working register 5 has the absolute address 101, and so on. The second group of working registers, r8-r15, has the offset values 0 to 7 respectively (i.e. r8 has the offset value 0, r9 has the offset value 1, and so on), this offset value being added to the value in Register Pointer 1.

For example, given that the value in Register Pointer 1 is 32, then working register 12 supplies an offset value of 4 (given by 12 minus 8) to the value in Register Pointer 1 to give an absolute address of 36.

Figure 2-6. Single 16 Register pointing Mode

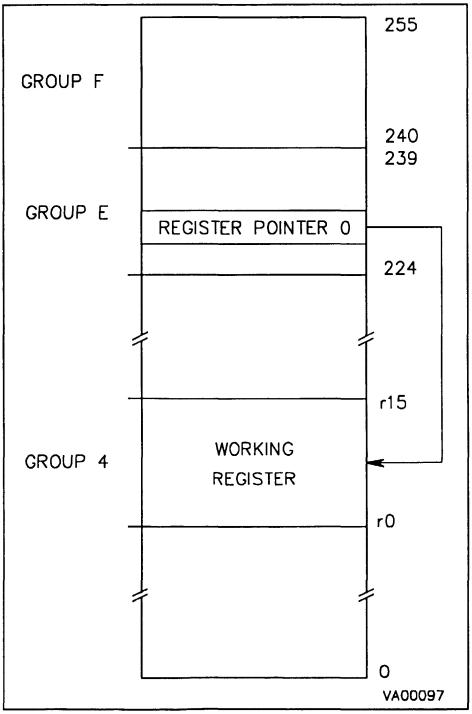
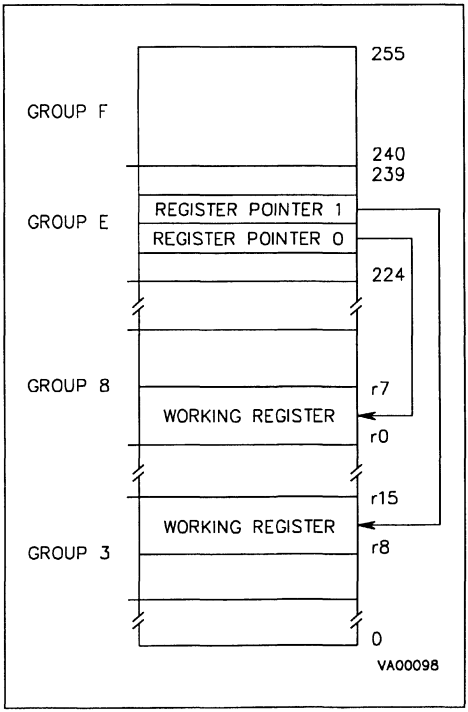


Figure 2-7. Double Register pointing Mode



SYSTEM REGISTERS (Continued)

2.3.4 Page Configuration

The pages are available to be used for the storage of control information (such as interrupt vector pointers) relevant to particular peripherals. There are up to 64 pages (each with 16 registers) based on registers R240-R255. These paged registers are addressable via the page pointer register (PPR), which is system register R234.

To address a paged register the page pointer register (R234) must be loaded with the relevant page number using the `spp` instruction (Set Page Pointer) and subsequently any address from the top (F) group (R240-R255) will be referred to that page.

For example if register 23 contains the value 44, the following sequence loads the third register R242 on page 5 with the value 44.

```
spp 5
ld R242, R23
```

PPR R234 (EAh) System Read/Write Page Pointer Register

Reset value : undefined

7							0
PP7	PP6	PP5	PP4	PP3	PP2	D1	D0

b7-b2 = **PP7-PP2:** *Page Pointer.* These bits contain the number (between 0 to 63) of the page chosen by the instruction `ssp` (Set Page Pointer). PP7 is the MSB of the page address. Once the page pointer has been set, there is no need to refresh it unless a different page is required.

b1-b0 = **D1,D0:** These bits are fixed by hardware to zero and are not affected by any writing instruction trying to modify their value.

PAGE 0 contains the control registers of:

- the external interrupt
- the watchdog timer
- the wait logic states
- the serial peripheral interface (SPI)

2.3.5 Mode Registers

This register MODER is located in the System Register Group at the address 235.

Using this register it is possible:

- to select either internal or external System and User Stack area,
- to manage the clock frequency
- to enable the Bus request and Wait signals when interfacing external memory.

MODER R235 (EBh) System Read/Write Mode Register

Reset value : 1110 0000

7							0
SSP	USP	DIV2	PRS2	PRS1	PRS0	BRQEN	HIMP

b7 = **SSP:** *System Stack Pointer.* This bit selects internal (in the Register File) or external (in the external Data Memory) System Stack area, logical "1" for internal, and logical "0" for external. After Reset the value of this bit is "1".

b6 = **USP:** *User Stack Pointer.* Same as bit 7 for the User Stack Pointer;

b5 = **DIV2:** *OSCIN Clock Divided by 2.* This bit controls the divide by 2 circuit which operates on the OSCIN Clock. A logical "1" value means that the OSCIN clock is internally divided by 2, and a logical "0" value means that no division of the OSCIN Clock occurs.

b4-b2 = **PRS2-PRS0:** *ST9 CPUCLK Prescaler.* These bits load the prescaling module of the internal clock (INTCLK). The prescaling value selects the frequency of the ST9 clock, which can be divided by 1 to 8. See Clock chapter for more information.

b1 = **BRQEN:** *Bus Request Enable.* This bit is a software enable of an External Bus Request. When set to "1", it enables a Bus Request on the BUSREQ pin.

b0 = **HIMP:** *High Impedance Enable.* When Port 0 and/or Port 1 are programmed as multiplexed address and Data lines to interface external Program and/or Data Memory, these lines and the Memory interface control lines ($\overline{AS}$, $\overline{DS}$, R/$\overline{W}$) can be forced into the High Impedance state by setting to "1" the HIMP bit. When this bit is reset, it has no effect on P0 and P1 lines.

If Port 1 is declared as an address AND as an I/O port (example: P10 ... P14 = Address, and P15 ... P17 = I/O), HIMP has no effect on the I/O lines (in the previous example: P15 ... P17).

SYSTEM REGISTERS (Continued)

2.3.6 Stack Pointers

There are two separate, double register stack pointers available (named System Stack Pointer and User Stack Pointer), both of which can address registers or memory.

The stack pointers point to the bottom of the stacks which are filled using the `push` commands and emptied using the `pop` commands. The stack pointer is automatically pre-decremented when data is "pushed in" and post-incremented when data is "popped out".

For example, the register address space is selected for a stack and the corresponding stack pointer register contains 220. When a byte of data is "pushed" into the stack, the stack pointer register is decremented to 219, then the data byte is "loaded" into register 219. Conversely, if a stack pointer register contains 189 and a byte of data is "popped" out, the byte of data is then extracted from the stack and then the stack pointer register is incremented to 190.

The `push` and `pop` commands used to manage the system stack area are made applicable to the user stack by adding the suffix U, while to use a stack instruction for a word a W is added.

For example `push` inserts data into the system stack, but an added U indicates the user stack and W means a word, so the instruction `pushuw` loads a word into the bottom of the user stack.

If the User Stack Pointer register contains 223 (working in register space) the instruction `pushuw` will decrement User Stack Pointer register to 222 and then load a word into register R222 and R221.

When bytes (or words) are "popped out" the values in those registers are left unchanged until fresh data is loaded into those locations. Thus when data is "popped" out from a stack area, the stack content remains unchanged.

Note. Stacks must not be located in the pages or the system register area.

The System Stack area and The System Stack Pointer

The System Stack area is used for the storage of temporarily suspended system and/or control registers, i.e. the Flag register and the Program counter, while interrupts are being serviced. For subroutine execution only the Program Counter needs to be saved in the System stack area.

There are two situations when this occurs automatically, one being when an interrupt occurs and the other when the instruction call subroutine is used. When the system stack area is in the Register File, the stack pointer, which points to the bottom of the stack, only needs one byte for addressing, in which case the System Stack Pointer Low Register (R239) is sufficient for addressing purposes. As a result the System Stack Pointer High Register (R238) becomes redundant BUT must be considered as reserved (please refer also to "spurious" memory access section). Clearly when the stack is external a full word address is necessary and so both registers are used to point, the even register providing the MSB and the odd register providing the LSB.

The User Stack area and User Stack Pointer

The User Stack area is completely free from all interference from automatic operations and so it provides a totally user controlled stacking area, that area being in any part of the memory which is of a RAM nature, or the first 14 groups of the general Register File i.e. not in the System register or Paged group.

The User Stack Pointer consists of two registers, R236 and R237, which are both used for addressing an external stack, while, when stacking in the Register File, the User Stack Pointer High Register, R236, becomes redundant but must be considered as reserved.

SYSTEM REGISTERS (Continued)

Stack location

Care is necessary when managing stacks as there is no limit to stack sizes apart from the bottom of any address space in which the stack is placed. Consequently programmers are advised to use a stack pointer value as high as possible, particularly when using the Register File as a stacking area. This will also benefit programmers who may locate the stacks in group D using, for example the instruction `ld R237, #223` which loads the value 223 into the User Stack Pointer Low Register. The Programmer will not need to remember to set the Register Pointer to 208 to gain access to registers in the D-group, a problem outlined in Register Pointing Techniques paragraph.

Stacks may be located anywhere in the first 14 groups of the Register File (internal stacks) or the data memory (external stacks). It is not necessary to set the data memory using the instruction `sdm` as external stack instructions automatically use the data memory.

Figure 2-8. System and/or User Stack in Register Stack Mode

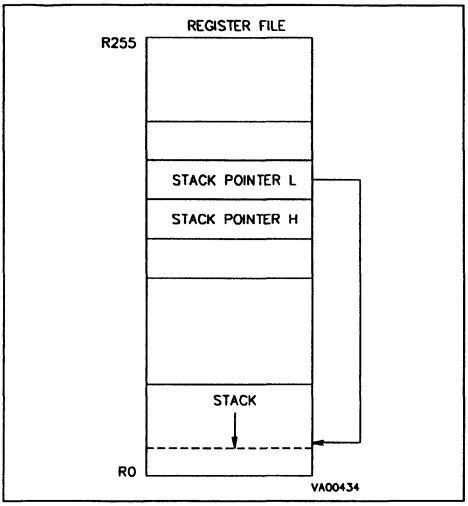
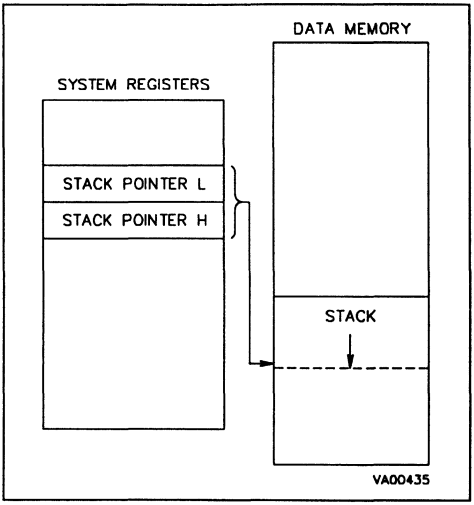


Figure 2-9. System and/or User Stack in Memory Stack Mode



USP R236 (ECh) Read/Write
User Stack Pointer High Byte

Reset value: undefined

7							0
USP15	USP14	USP13	USP12	USP11	USP10	USP9	USP8

SSP R238 (EEh) Read/Write
System Stack Pointer High Byte

Reset value: undefined

7							0
SSP15	SSP14	SSP13	SSP12	SSP11	SSP10	SSP9	SSP8

USP R237 (EDh) Read/Write
User Stack Pointer Low Byte

Reset value: undefined

7							0
USP7	USP6	USP5	USP4	USP3	USP2	USP1	USP0

SSP R239 (EFh) Read/Write
System Stack Pointer Low Byte

Reset value: undefined

7							0
SSP7	SSP6	SSP5	SSP4	SSP3	SSP2	SSP1	SSP0

NOTES

3 MEMORY

3.1 INTRODUCTION

The memory of the ST9293 is functionally divided into two areas, the Register File and Memory. The Memory may optionally be divided into two spaces, each having a maximum of 65,536 bytes. The two memory spaces are separated by function, one space for Program code, the other for Data.
The ST9293 16K to 48K bytes of on-chip ROM memory are selected in the PROGRAM space.
The 768/640 bytes of on-chip RAM are mapped into both Program and Data space.

The memory spaces are selected by the execution of the SDM and SPM instructions (Set Data Memory and Set Program Memory, respectively). There is no need to use either of these instructions again until the memory area required is to be changed. This requirement is not necessary when using the memory indirect to memory indirect post-increment addressing mode (the memory types are specified in the instructions: LDPP, LDPD, LDDP, LDDD).

Either the Data Memory or the Program Memory can be addressed using any of the memory addressing modes.

3.1.1 Program Space

The Program memory space of the ST9293 consists of 32K or 48K bytes of on-chip ROM memory.

The first 256 memory locations from address 0 to 0FFh (hexadecimal) hold the Reset Vector, the Top-Level (Pseudo Non-Maskable) interrupt, the Divide by Zero Trap vector and, optionally, the interrupt vector table for use with the on-chip peripherals and the external interrupt sources. Each vector is contained in two consecutive byte locations, the high order address held in the lower (even) byte, the low order address held in the upper (odd) byte, forming the address which is loaded into the Program Counter when selected by the interrupt vector provided by the interrupt source. This should point to the relevant Interrupt Service routine provided by the User for immediate response to the interrupt.

3.1.2 Data Space

The ST9293 addresses the 640 bytes of on-chip RAM memory from addresses 8000 to 827Fh in both Program and Data Space. On-chip general purpose Registers may be used as additional RAM memory for minimum chip count systems.

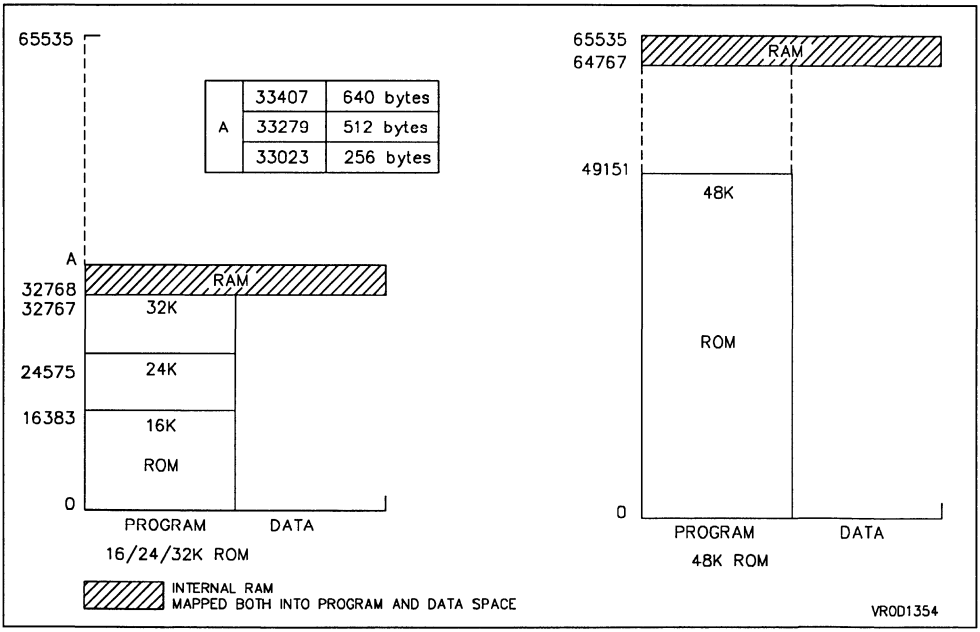
The Data Space is selected by the execution of the SDM instruction. All subsequent memory references will access the Data Space. When a separate Data Space is not required, data may stored in RAM or ROM memory within the Program Space.

MEMORY (Continued)

Table 3-1. ROM and RAM Address Configuration

Device Suffix	ROM Size (Bytes)	ROM Addresses		RAM Size (Bytes)	RAM Addresses	
J7	48K	0 - 49151	dec	768	64767 - 65535	dec
		0000 - BFFF	hex		FD00 - FFFF	hex
J5	32K	0 - 32767	dec	640	32768 - 33407	dec
		0000 - 7FFF	hex		8000 - 827F	hex
J3	24K	0 - 24575	dec	512	32768 - 33279	dec
		0000 - 5FFF	hex		8000 - 81FF	hex
J1	16K	0 - 16383	dec	256	32768 - 33023	dec
		0000 - 3FFF	hex		8000 - 80FF	hex

Figure 3-1. Memory MAP



4 INTERRUPTS

4.1 INTRODUCTION

The ST9 responds to peripheral events and external events through its Interrupt channels. When such an event occurs, if previously enabled and according to a priority mechanism, the current program execution can be suspended to allow the ST9 to execute a specific response routine. If the event generates an interrupt request, the current program status is saved after the current instruction is completed and the CPU control passes to the Interrupt Service Routine.

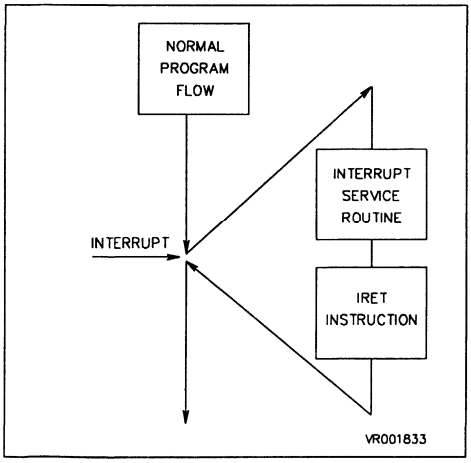
The ST9 CPU can receive requests from the following type of sources:

- On-chip peripherals
- External pins
- Top-Level Pseudo-non-maskable interrupt

According to the on-chip peripheral features, an event occurrence can generate an Interrupt request depending on the selected mode.

Up to eight external interrupt channels, with programmable input trigger edge, are available. In addition, a dedicated interrupt channel, set to the Top-level priority, can be devoted either to the external pin NMI (to provide a Non-Maskable-Interrupt) or to the Timer/Watchdog. Interrupt service routines are addressed through a vector table mapped in Program Memory.

Figure 4-1. Interrupt Flow



4.2 INTERRUPT VECTORIZATION

The ST9 implements an interrupt vectoring structure that allows the on-chip peripheral to identify the location of the first instruction of the Interrupt Service Routine (IVR) automatically.

When the interrupt request is acknowledged, the peripheral interrupt module provides, through its Interrupt Vector Register (IVR), a vector to point into the vector table of locations containing the start addresses of the Interrupt Service Routines (defined by the programmer).

Each peripheral has a specific IVR mapped within its Register File pages.

The Interrupt Vector table, containing the list of the addresses of the Interrupt Service Routines, is located in the first 256 locations of the Program Memory. The first 6 locations of the Program Memory are reserved for:

Address Content

0	Address high of Power on Reset routine
1	Address low of Power on Reset routine
2	Address high of Divide by zero trap Subroutine
3	Address low of Divide by zero trap Subroutine
4	Address high of Top Level Interrupt routine
5	Address low of Top Level Interrupt routine

With one Interrupt Vector register, it is possible to address more interrupt service routines; in fact, several peripherals share the same interrupt vector register among several interrupt channels. The most significant bits of the vector are user programmable to define the base vector address inside the vector table in the program memory, the least significant bits are controlled by the interrupt module in hardware to select the specific vector.

Note: The first 256 locations of the program memory can contain program code. Other than the Reset vector, they are not exclusively reserved to the vector table.

Warning. *Although the Divide by Zero Trap operates as an interrupt, the FLAG Register is not pushed onto the system Stack automatically. As a result it must be regarded as a subroutine, and the acknowledge routine must end with the* RET *instruction.*

INTERRUPT VECTORIZATION (Continued)

Figure 4-2. Vectors and Associated Routines

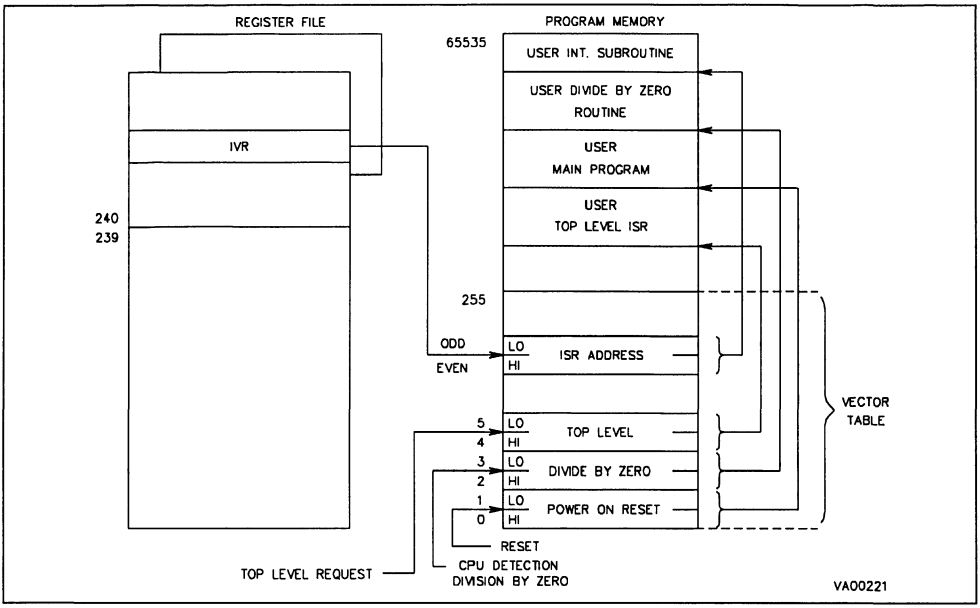
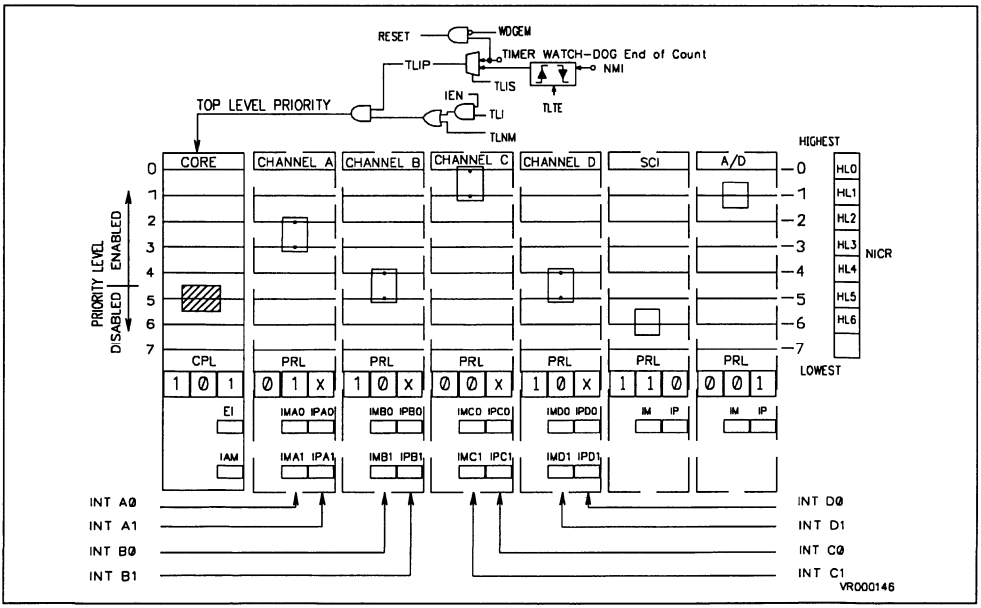


Figure 4-3. Interrupt Architecture, Example of priority Allocations



4.3 INTERRUPT PRIORITY LEVEL ARCHITECTURE

The ST9 supports a fully programmable interrupt priority structure. Figure 4-4 shows a conceptual description.

9 priority levels are available to define the channel priority relationship. Each channel has a 3 bit field, PRL (Priority Level), that defines its priority level among 8 programmable levels. The ninth level (Top Level Priority) is reserved for the Timer/Watchdog or the External Pseudo Non-Maskable Interrupt. The On-chip peripheral channel and the eight external interrupt sources can be programmed within eight priority levels: level 7 has the lowest priority, level 0 has the highest priority.

If several units are located at the same priority level, an internal daisy chain, fixed for each ST9 device, defines the priority relationship within that level.

The PRL bits are used to define the priority level for interrupt requests.

Top level priority interrupt (highest) can be assigned either to the external Pseudo Non-Maskable interrupt or to the internal Timer/Watch-Dog. An Interrupt service routine at this level cannot be interrupted in any arbitration mode. Its mask can be both maskable (TLI) or non-maskable (TLNM).

4.4 PRIORITY LEVEL ARBITRATION

The 3 bits of CPL (Current Priority Level) in the Central Interrupt Control Register contain the priority of the currently running program (CPU priority). CPL is set to 7 (lowest priority) upon reset and can be modified during program execution either by software or automatically by hardware according to the selected Arbitration Mode.

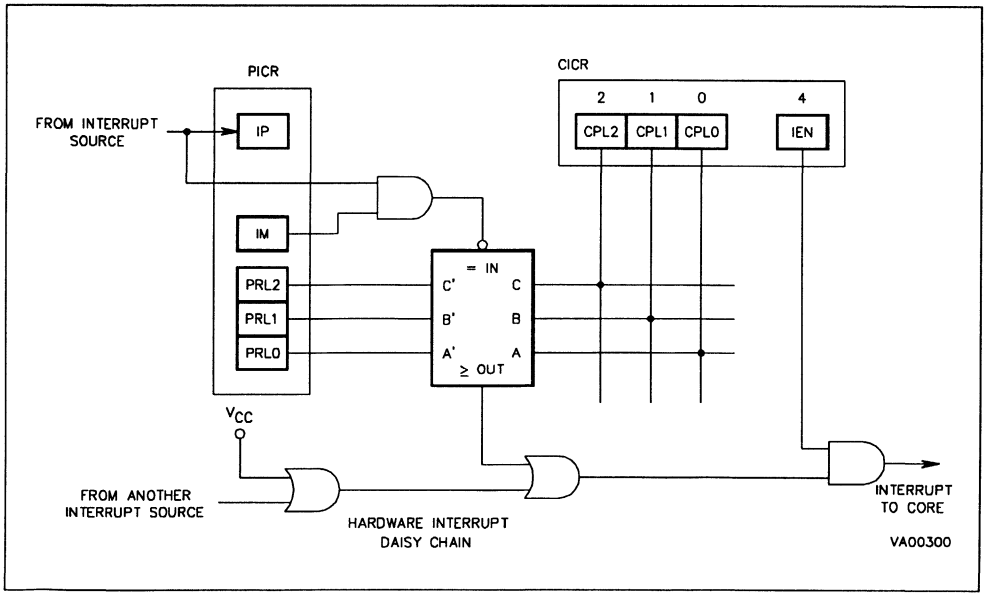
During every instruction an arbitration phase is made between every channel capable of generating an Interrupt, each priority level is compared to all the other requests. If the highest priority request is an interrupt, it must be *higher* than the CPL value in order to be acknowledged.

The priority of the Top Level Interrupt overrides every other priority.

If two or more requests occur at the same instant of time and at the same priority level, an on-chip daisy chain, specific to every ST9 version, selects the channel with the highest position in the chain. The position in the chain is shown in table 4-1.

ST9 provides two interrupt arbitration modes: Concurrent and Nested modes. The Concurrent mode is the standard interrupt arbitration mode while the Nested mode improves the effective interrupt response time when a nesting of the service routines is required according to the request priority levels.

Figure 4-4. Interrupt Logic



PRIORITY LEVEL ARBITRATION (Continued)

The control bit IAM (CICR.3) selects the Concurrent Arbitration mode (when reset to "0") or the Nested Arbitration Mode (when set to "1").

Table 4-1. Daisy Chain Priorities

Applicable for ST9293

Highest Position	INTA0
	INTA1
	INTB0
	INTB1
	INTC0
	INTC1
	INTD0
Lowest Position	INTD1

***Warning**: Although the divide by Zero Trap operates as an interrupt, the FLAG Register is not pushed onto the system Stack automatically. As a result it must be regarded as a subroutine, and the acknowledge routine must end with the RET instruction.*

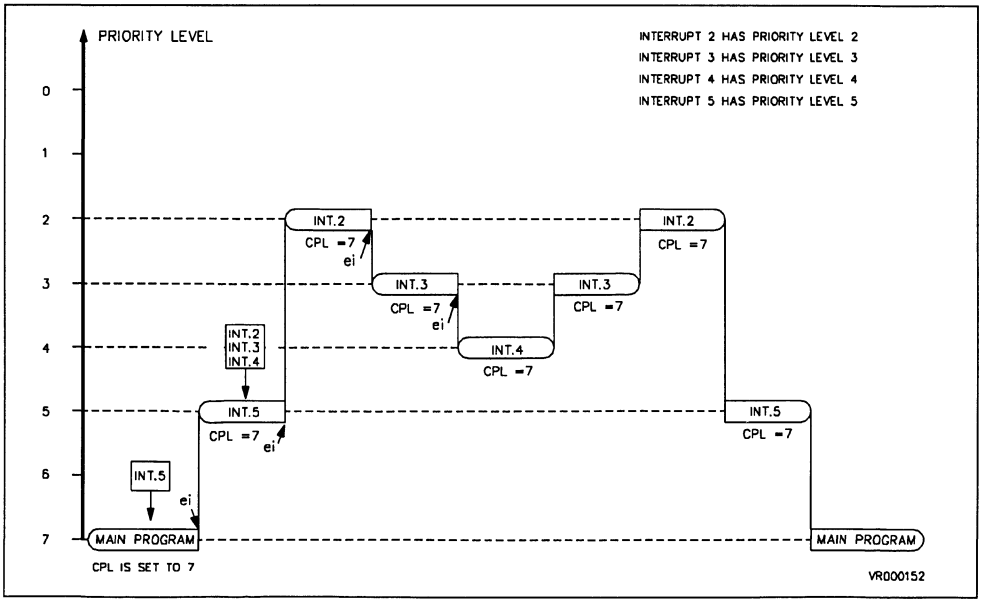
4.4.1 Concurrent Mode

This mode is selected when the IAM bit is cleared (reset condition). The arbitration phase, performed during every instruction, selects the request with the highest priority level.

If the highest priority request is an interrupt request and its priority value is higher than the Current Priority Value CICR.2,1,0 (R230.2,1,0), the interrupt request will be acknowledged at the end of the current instruction. The interrupt Machine Cycle performs the following steps:

- 1. Disables all the maskable interrupt requests by clearing CICR.IEN
- 2. Pushes the PC low byte into the system stack
- 3. Pushes the PC high byte into the system stack
- 4. Pushes the Flag register into the system stack
- 5. Loads the PC with the 16-bit vector stored in the Vector Table, pointed to by the Interrupt Vector Register (IVR).

Figure 4-5. Example of a Sequence of Interrupt Requests with :
- Concurrent mode
- EI set to 1 during the interrupt routine execution



PRIORITY LEVEL ARBITRATION (Continued)

The Interrupt Service Routine must be concluded with the `iret` instruction. The `iret` instruction executes the following operations:

- 1. Pops off the Flag register from the system Stack
- 2. Pops off PC high byte from the system Stack
- 3. Pops off PC low byte from the system Stack
- 4. Enables all the un-masked Interrupts, by setting the CICR.IEN bit

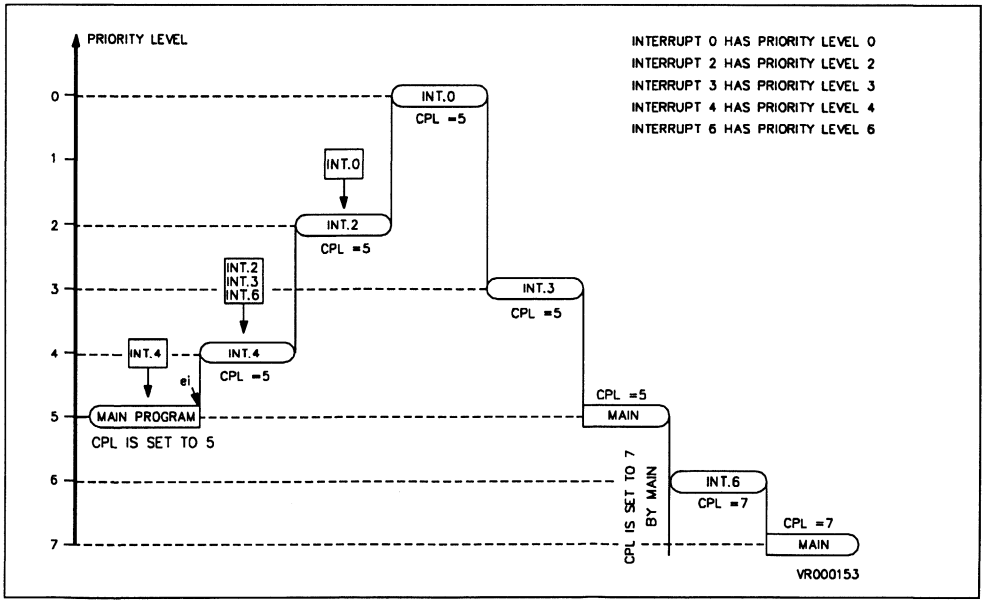
The suspended program execution is thus recovered at the interrupted instruction. All pending interrupts existing, or having occurred during the interrupt service routine execution, remain pending until the Enable Interrupt instruction (even if it is executed during the interrupt service routine).

NOTE: When Concurrent mode is selected, the source priority level is meaningful only during the arbitration phase, where it is compared to all the other priority levels and the CPL, but no trace is kept of its value during the Interrupt Service Routine. Therefore, if other requests are issued, once the global CICR.IEN is enabled again, they will be acknowledged regardless of the Interrupt Service Routine priority value; if no care is taken by the programmer, unpleasant side effects can take place.

A typical case is the following: 3 pending requests with different priority levels (ie 2,3,4) generate requests at the same time (because the associated events occurred during the same instruction). The three interrupt service routines set Interrupt Enable (IEN, CICR.4) by the `ei` instruction at the beginning of the routine to avoid a high interrupt response time to requests with a priority higher than the one under service (usually, the higher the priority, the sooner the routine must be executed). Unfortunately, what will happen in this case is that the three interrupt servicing routines will be executed exactly in the opposite order of their priority. Interrupt routine level 2 will be acknowledged first, then, when the `ei` instruction is executed, it will be interrupted by interrupt routine level 3, which itself will be interrupted by interrupt routine level 4. When interrupt routine level 4 is completed, interrupt routine level 3 will be recovered and finally, interrupt routine level 2.

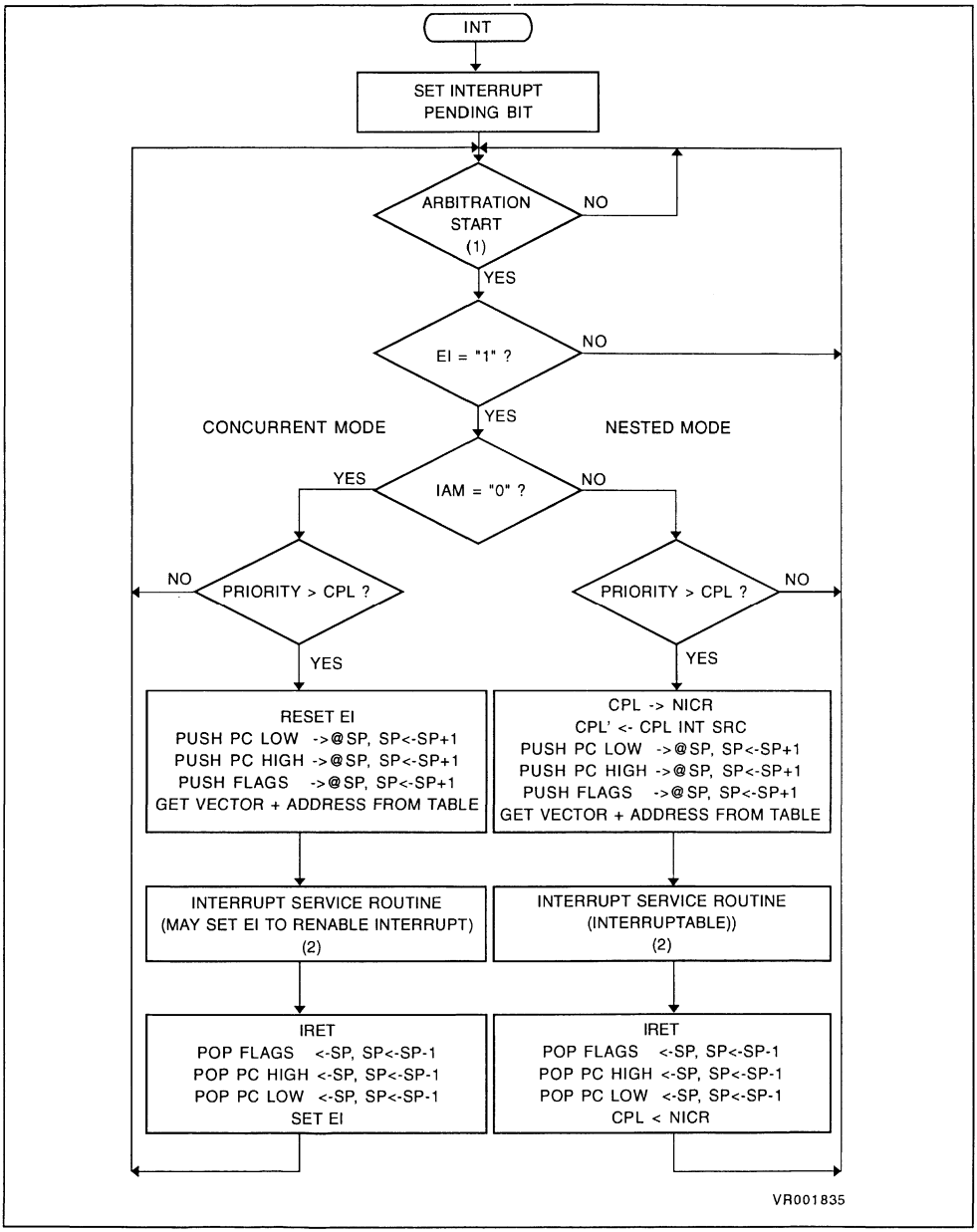
Therefore, **it is recommended, in concurrent mode, to avoid the insertion of the `ei` instruction in the interrupt subroutine**, which can trigger this LIFO (Last In, First Out) sequence of interrupt processing.

Figure 4-6. Example of a Sequence of Interrupt Requests with :
- Concurrent mode
- EI unchanged by the interrupt routines



PRIORITY LEVEL ARBITRATION (Continued)

Figure 4-7. Interrupt Mode Flow-Chart



Notes:
1. The interrupt arbitration starts 6 CPUCLK cycles before the end of execution of each instruction (5 cycles during WFI).
2. Clear interrupt pending bit

PRIORITY LEVEL ARBITRATION (Continued)

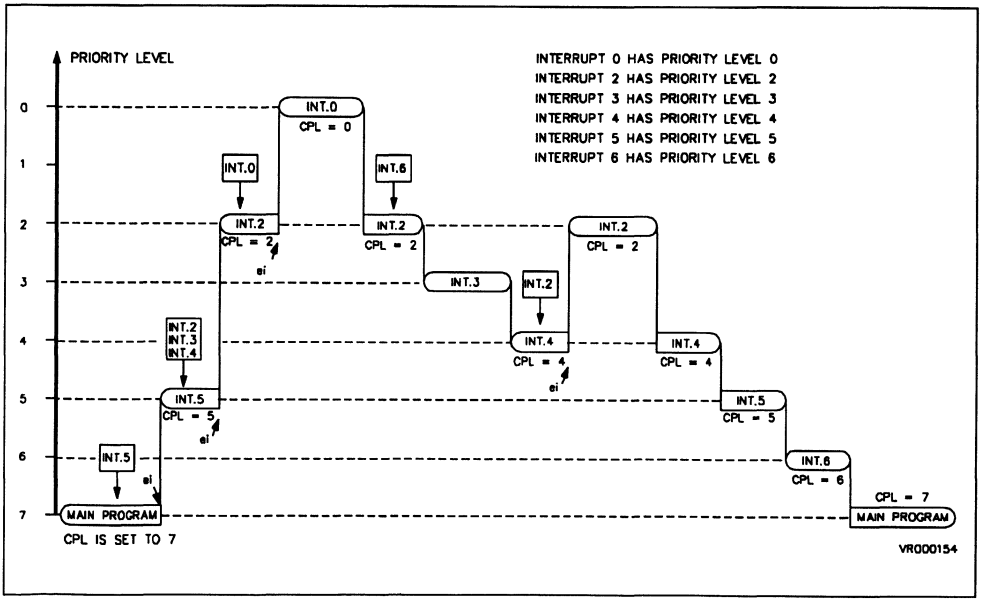
4.4.2 Nested Mode

The difference of the Nested mode to the Concurrent mode consists of the modification of the CPL value during the interrupt processing. The arbitration phase is basically identical to the concurrent Mode, however once the request is acknowledged, the current CPL value is saved in the Nested Interrupt Control Register (NICR, R247 page 0) by setting the NICR bit corresponding to the CPL value (i.e. if the CPL is 3, NICR.3 bit will be set). The CPL value is then updated with the Priority value of the request just acknowledged, in this way the next arbitration cycle will be performed against the priority level of the Service Routine in progress.

The Interrupt Machine Cycle will perform the following steps:

- Disable all the maskable interrupts by clearing IEN
- Save the CPL value into the special stack NICR to hold the priority level of the suspended routine
- Store in CPL the priority level of the acknowledged routine, so that the next request priority will be compared with the one of the routine under service
- Push the PC-low byte into the System Stack
- Push the PC-high byte into the System Stack
- Push the Flag Register into the System Stack

Figure 4-8. Example of a Sequence of Interrupt Requests with :
- Nested mode
- EI set to 1 during the interrupt routine execution



PRIORITY LEVEL ARBITRATION (Continued)

- Load the PC with the vector pointed by IVR.

The `iret` Interrupt Return instruction executes the following steps:

- 1. Pop off the Flag Register from the System Stack
- 2. Pop off the PC-high byte from the System Stack
- 3. Pop off the PC-low byte from the System Stack
- 4. Enable all the unmasked interrupts by setting the IEN bit
- 5. Recover the interrupted routine priority level by popping the value from the special register (NICR) and by copying it into CPL.

The suspended execution is thus recovered at the interrupted instruction.

REMARKS

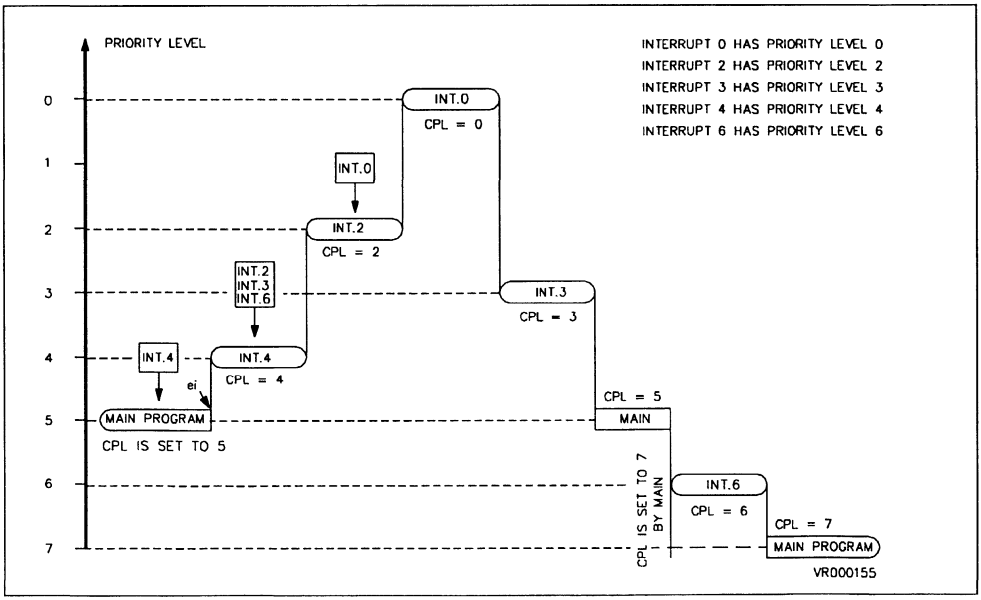
1) Dynamic priority level modification: the main program and routines can be specifically prioritized. Since CPL is represented by 3 bits in a read/write register, it is possible to modify dynamically the current priority value during the program execution. This means that a critical section can have a higher priority with respect to other interrupt requests. Furthermore it is possible to prioritize even the Main Program execution by modifying CPL during its execution.

2) Maximum number of nestings: No more than 8 routines can be nested. If an interrupt routine at level N is being serviced, no other Interrupts located at level N can interrupt it. This guarantees a maximum number of 8 nested levels including the Top Level Interrupt request.

3) Priority level 7: Interrupt requests at level 7 cannot be acknowledged as their priority cannot be higher than the CPL value. This can be of use in a fully polled interrupt environment.

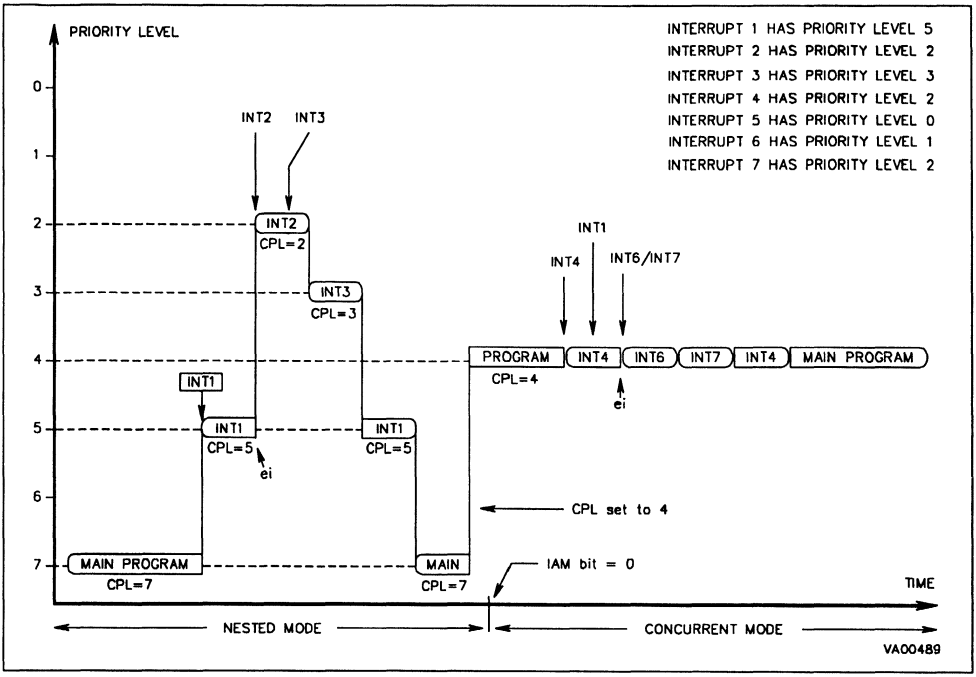
A nested/concurrent mode sequence is given on Figure 4-10. This example clearly shows that Nested and Concurrent modes are defined by the user. Note that here the Y axis is referenced by CPL, instead of the source priority level, and *that Interrupt 1 stays pending*, having a priority level lower than CPL.

Figure 4-9. Example of a Sequence of Interrupt Requests with :
- Nested mode
- EI unchanged by the interrupt routiness



PRIORITY LEVEL ARBITRATION (Continued)

Figure 4-10. Example of a Nested and Concurrent Mode Sequence



4.5 EXTERNAL INTERRUPTS

The standard ST9 core contains 8 external interrupts sources grouped into four pairs.

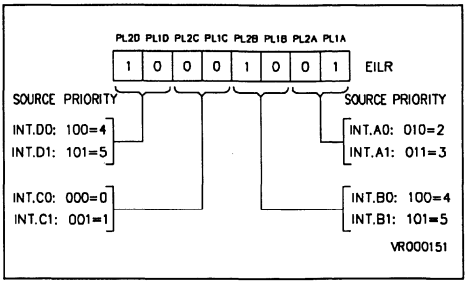
Table 4-2. External Interrupt Channel Grouping

External Interrupt	Channel
INT7 INT6	INTD1 INTD0
INT5 INT4	INTC1 INTC0
INT3 INT2	INTB1 INTB0
INT1 INT0	INTA1 INTA0

Each source has a trigger control bit TEA0,..TED1 (R242,EITR.0,..,7 Page 0) to select triggering on the rising or falling edge of the external pin. If the Trigger control bit is set to "1", the corresponding pending bit IPA0,..,IPD1 (R243,EIPR.0,..,7 Page 0) is set on the input pin rising edge, if it is cleared, the pending bit is set on the falling edge of the input pin. Each source can be individually masked through the corresponding control bit IMA0,..,IMD1 (EIMR.7,..,0). See Figure 4-12.

The priority level of the external interrupt sources can be programmed among the eight priority levels with the control register EIPLR (R245). The priority level of each pair is software defined using the bits PRL2,PRL1. For each pair, the even channel (A0,B0,C0,D0) of the group has the even priority level and the odd channel (A1,B1,C1,D1) has the odd (lower) priority level. Figure 4-11 shows an example of priority levels.

Figure 4-11. Priority Level Examples



- The source of the interrupt channel A0 can be selected between the external pin INT0 (when IA0S = "1", the reset value) or the On-chip Timer/Watchdog peripheral (when IA0S = "0").
- The source of the interrupt channel B0 can be selected between the external pin INT2 (when (SPEN,BMS)=(0,0)) or the on-chip SPI peripheral.

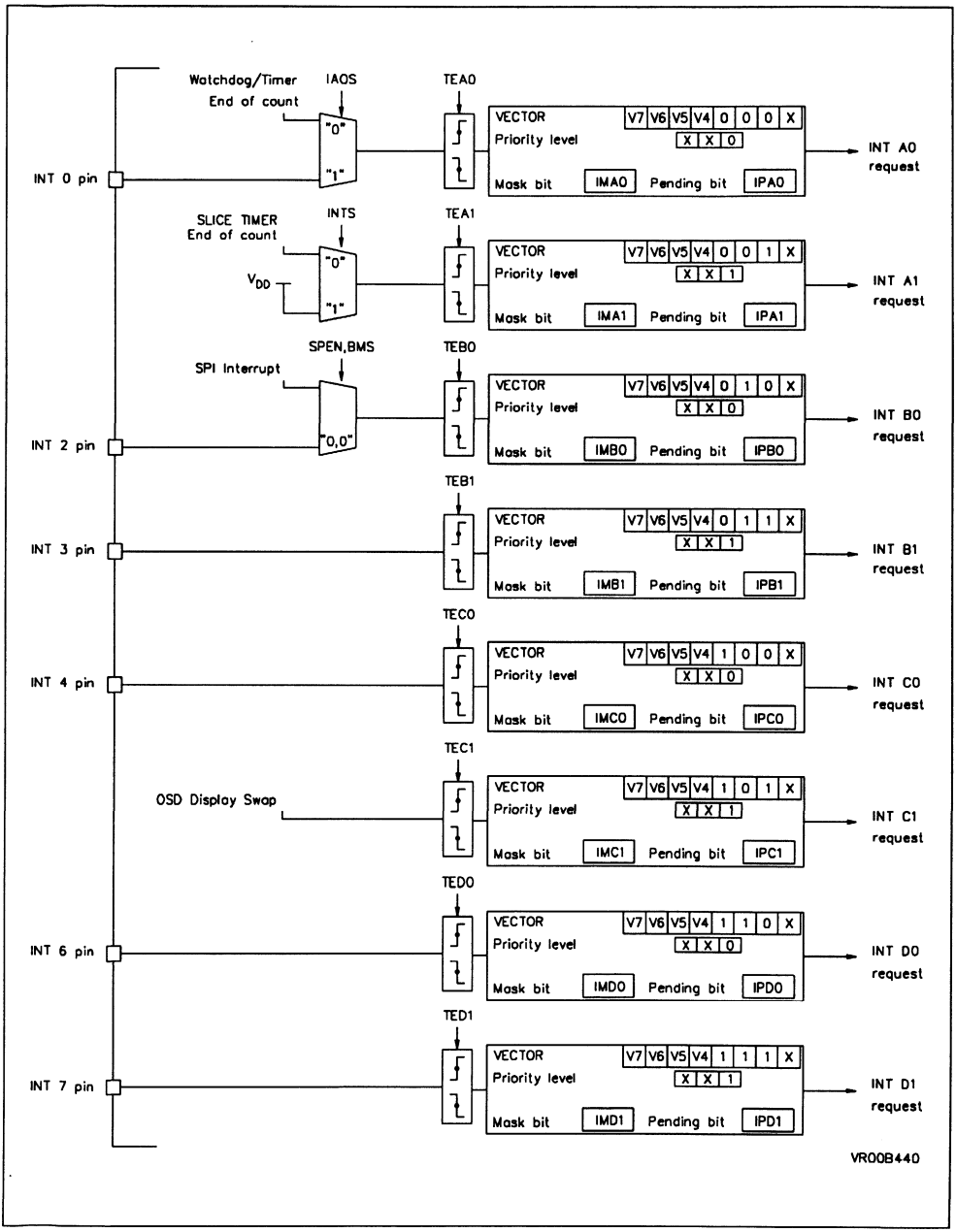
All other interrupt channels have an input pin as source, however, the input line may be multiplexed with an on-chip peripheral I/O or connected to an input pin that performs also other function (as in the case of the handshake feature).

Table 4-3. ST9293 External Interrupt Source Selection

INT0	INTA0
Timer/Watchdog End of Count	
INT2	INTB0
SPI Interrupt	
Slice Timer End of Count	INTA1
OSD Display Swap	INTC1

EXTERNAL INTERRUPTS (Continued)

Figure 4-12. External Interrupts Control Bits and Vectors



4.6 TOP LEVEL INTERRUPT

The Top Level Interrupt channel can be assigned either to the external pin NMI or to the Timer/Watchdog according to the status of the control bit EIVR.TLIS (R246.2, Page 0). If this bit is high (the reset condition) the source is the external pin NMI, if it is low, the source is the Timer/ Watchdog End Of Count. When the source is the NMI external pin, the control bit EIVR.TLTEV (R246.3; Page 0) selects between the rising (if set) or falling (if cleared) edge generating the interrupt request. When the selected event occurs, the CICR.TLIP bit (R230.6) is set. Depending on the mask situation, a Top Level Interrupt request may be generated. Two kinds of masks are available, a Maskable mask and a Non-Maskable mask. The first mask is the bit CICR.TLI (R230.5): it can be set or cleared to enable or disable respectively the Top Level Interrupt request. If it is enabled, the global Enable Interrupt bit CICR.IEN (R230.4) must also be enabled in order to allow a Top Level Request.

The second mask NICR.TLNM (R247.7) is a set-only mask. Once set, it enables the Top Level Interrupt request independently of the value of CICR.IEN and it cannot be cleared by program. Only the processor RESET cycle can clear this bit.

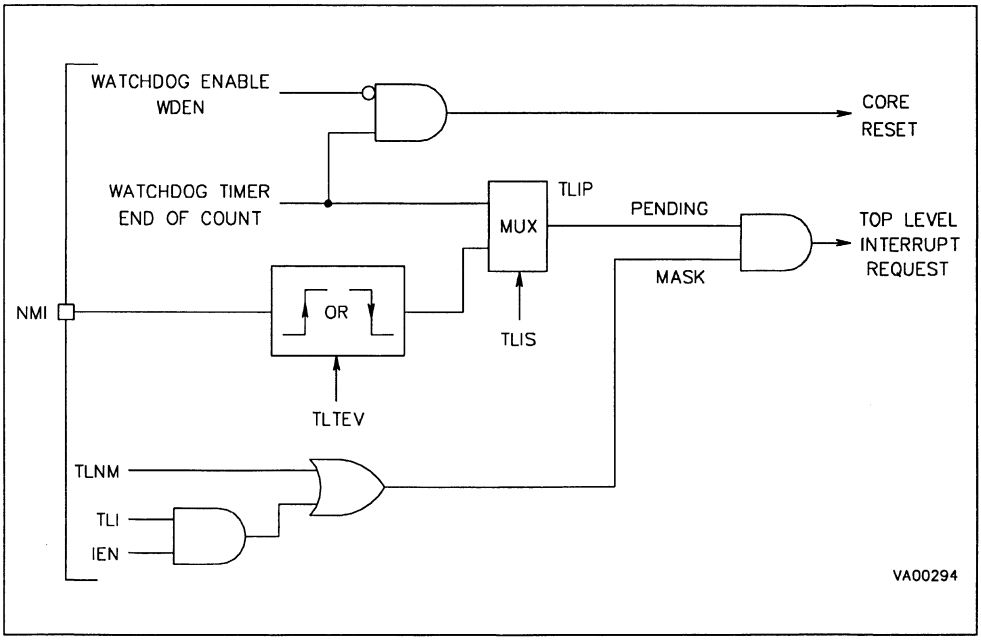
The Top Level Interrupt Service Routine cannot be interrupted by any other interrupt request, in any arbitration mode, even by another Top Level Interrupt request.

Warning. *The interrupt machine cycle of the Top Level Interrupt does not clear the CICR.IEN bit, and the corresponding* `iret` *does not set it.*

4.7 ON-CHIP PERIPHERAL INTERRUPTS

The general structure of the peripheral interrupt unit is described here, however each on-chip peripheral has its own specific interrupt unit containing one or more interrupt channels, or DMA channels. Please refer to the specific peripheral chapter for the description of its interrupt features and control registers.

Figure 4-13. Top Level Interrupt Structure

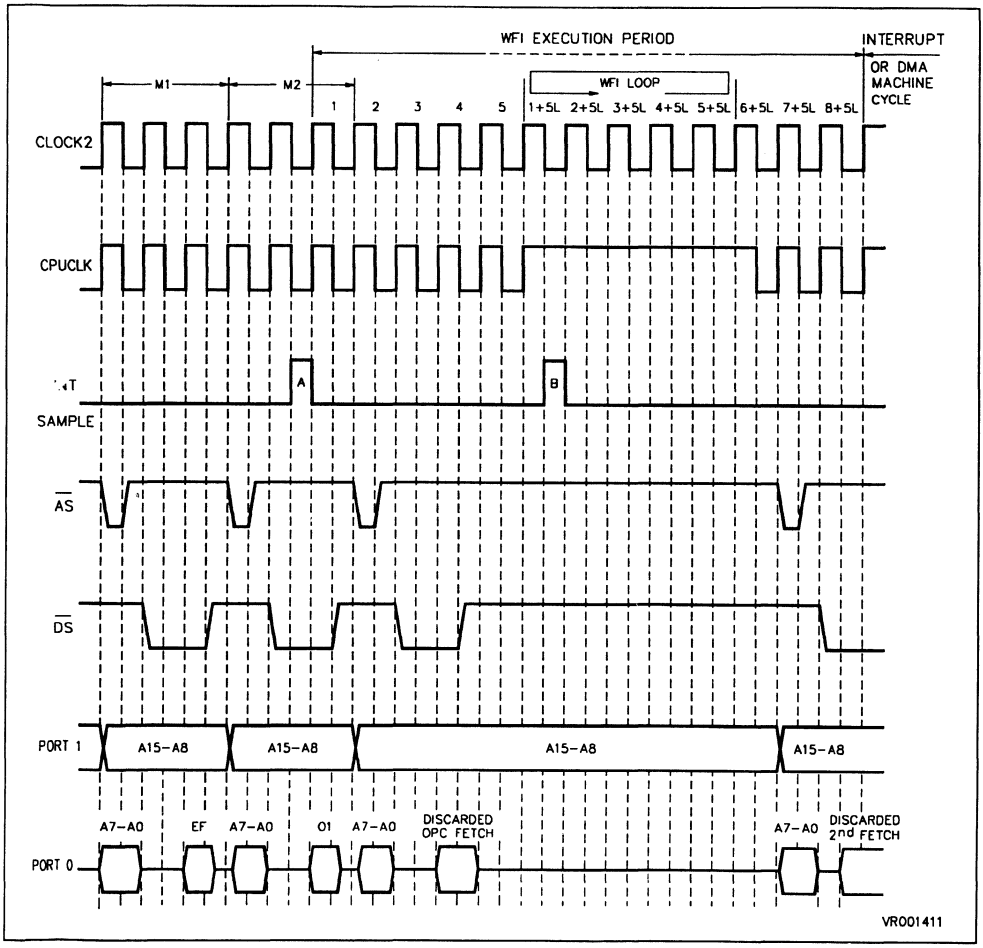


ON-CHIP PERIPHERAL INTERRUPTS (Continued)

The on-chip peripheral interrupt channels provide the following control bits:

- Interrupt Pending bit (IP)
  Set by hardware when the Trigger Event occurs. Can be set/cleared by software to generate/cancel pending interrupts and give the status for Interrupt polling.
- Interrupt Mask bit (IM)
  If IM = "0", no interrupt request is generated. If IM ="1" an interrupt request is generated whenever IP = "1" and CICR.IEN = "1".
- Priority Level (PRL, 3 bits)
  These bits define the source priority level
  PRL=0: the highest priority
  PRL=7: the lowest priority (the interrupt cannot be acknowledged)
- Interrupt Vector Register (IVR, up to 7 bits)
  The IVR points to the vector table which itself contains the interrupt routine start address.

Figure 4-14. Wait For Interrupt Timing



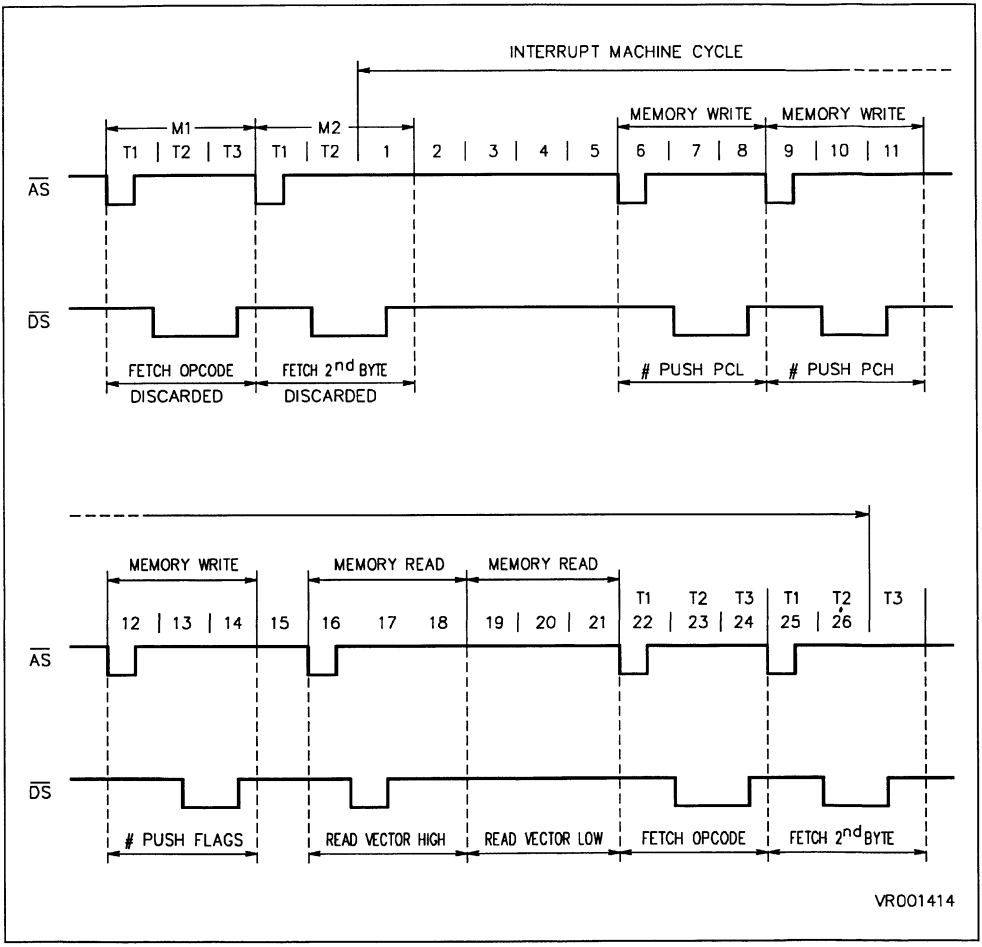
4.8 WAIT FOR INTERRUPT INSTRUCTION

The Wait For Interrupt instruction suspends program execution until an interrupt request is acknowledged. During the WFI instruction, the CPUCLK is halted while INTCLK keeps running. Under this state, the power consumption of the processor is lowered by the CORE power consumption value.

4.9 INTERRUPT RESPONSE TIME

Interrupt requests are sampled 6 CPUCLK cycles before the end of the instruction. If Wait For Interrupt is in progress, requests are sampled every 5 CPUCLK cycles. If the interrupt request comes from an external pin, the programmed event has to be set a minimum of one CPUCLK cycle before the sampling time.

Figure 4-15. Interrupt Acknowledge Timing



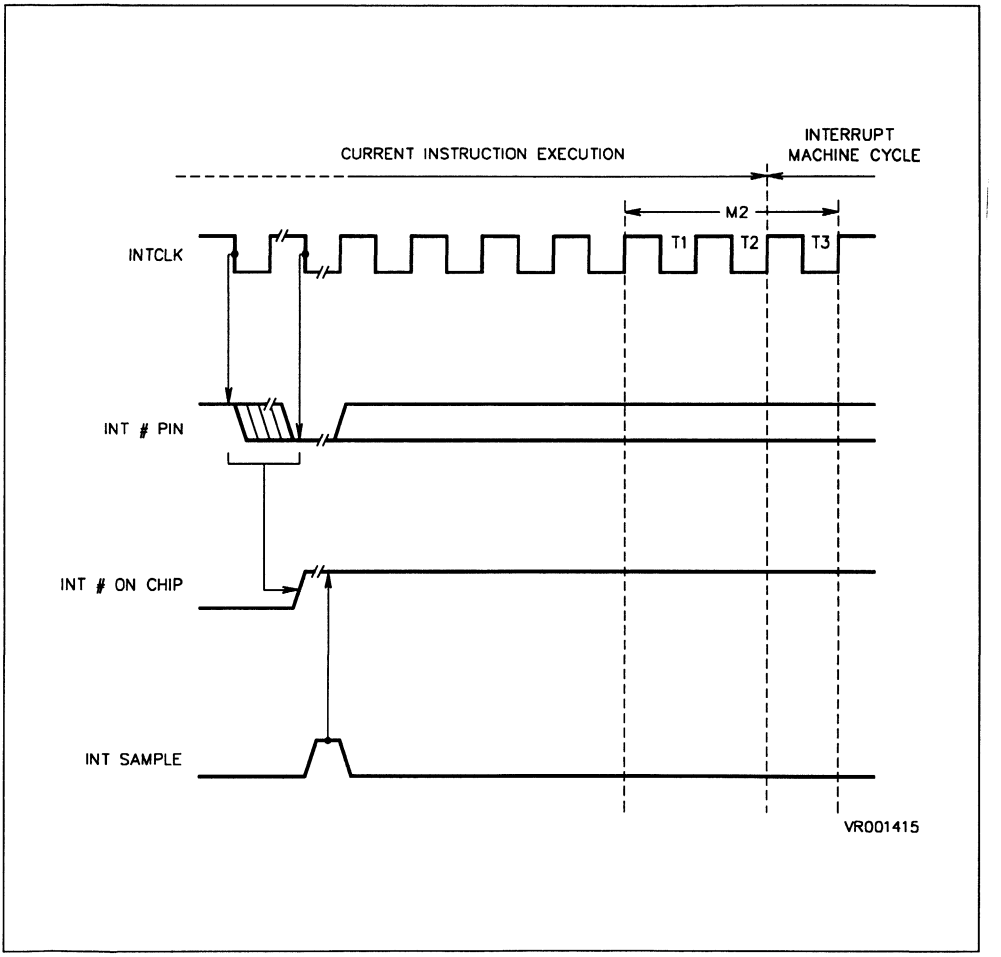
INTERRUPT RESPONSE TIME (Continued)

In order to guarantee the falling/rising edge detection, input signals must be kept low/high for a minimum of one CPUCLK cycle.

An interrupt machine cycle takes 26 internal clock cycles (CPUCLK), with some exceptions as follows:

- 28 internal clock cycles (CPUCLK), if a Wait For Interrupt is in progress
- 32 internal clock cycles (CPUCLK), if the acknowledge cycle follows a DMA transfer with Register File

Figure 4-16. External Interrupt Response Time



4.10 INTERRUPT REGISTERS

CICR R230 (E6h) System Read/Write
Central Interrupt Control Register

Reset value: 1000 0111 (87h)

7							0
GCEN	TLIP	TLI	IEN	IAM	CPL2	CPL1	CPL0

b7 = **GCEN:** *Global Counter Enable bit.* When set the 16 bit MultiFunction Timers are enabled (see Timer Control Register in MULTI FUNCTION TIMER chapter)

b6 = **TLIP:** *Top Level Interrupt Pending bit.* Set by hardware when the Trigger Event occurs. Cleared by hardware when the Top Level Interrupt is acknowledged.

b5 = **TLI:** *Top Level Interrupt bit.* If TLI ="1", and IEN is set, a Top Level Interrupt request is generated as TLIP is set. If TLI = "0", a request is generated only if TLNM is set.

b4 = **IEN:** *Interrupt Enable.* If IEN = "0", no maskable Interrupt requests are generated. This bit is cleared by the interrupt machine cycle and it is set by the IRET instruction of maskable routines.

b3 = **IAM:** *Interrupt Arbitration Mode.* If IAM = "0", Concurrent Arbitration Mode is selected; If IAM = "1" Nested Mode is selected.

b2-b0 = **CPL2, CPL1, CPL0:** *Current Priority Level.* Defines the Current Priority Level under service. CPL=0 is the highest priority. CPL=7 is the lowest priority. This bits may be modified directly by the interrupt hardware when the Nested Interrupt Mode is used.

EITR R242 (F2h) Page 0 Read/Write
External Interrupt Trigger Event Register

Reset value: 0000 0000 (00h)

7							0
TED1	TED0	TEC1	TEC0	TEB1	TEB0	TEA1	TEA0

If TExy bit is set, the pending bit will be set upon the rising edge of the input signal.

If TExy is cleared, the pending bit will be set upon the falling edge of the input signal.

All bits are set/reset only by software.

b7 = **TED1:** *Trigger Event of Interrupt Channel D1*

b6 = **TED0:** *Trigger Event of Interrupt Channel D0*

b5 = **TEC1:** *Trigger Event of Interrupt Channel C1*

b4 = **TEC0:** *Trigger Event of Interrupt Channel C0*

b3 = **TEB1:** *Trigger Event of Interrupt Channel B1*

b2 = **TEB0:** *Trigger Event of Interrupt Channel B0*

b1 = **TEA1:** *Trigger Event of Interrupt Channel A1*

b0 = **TEA0:** *Trigger Event of Interrupt Channel A0*

IDPR R243 (F3h) Page 0 Read/Write
External Interrupt Pending Register

Reset value: 0000 0000 (00h)

7							0
IPD1	IPD0	IPC1	IPC0	IPB1	IPB0	IPA1	IPA0

b7 = **IPD1:** *Interrupt Pending bit Channel D1*

b6 = **IPD0:** *Interrupt Pending bit Channel D0*

b5 = **IPC1:** *Interrupt Pending bit Channel C1*

b4 = **IPC0:** *Interrupt Pending bit Channel C0*

b3 = **IPB1:** *Interrupt Pending bit Channel B1*

b2 = **IPB0:** *Interrupt Pending bit Channel B0*

b1 = **IPA1:** *Interrupt Pending bit Channel A1*

b0 = **IPA0:** *Interrupt Pending bit Channel A0*

IP bits are hardware set upon the occurence of the trigger event and are reset by the interrupt acknowledge machine cycle.

Note. IP bits may be set by the programmer to implement a software interrupt.

INTERRUPT REGISTERS (Continued)

EIMR R244 (F4h) Page 0 Read/Write
External Interrupt Mask-bit Register

Reset value: 0000 0000 (00h)

7							0
IMD1	IMD0	IMC1	IMC0	IMB1	IMB0	IMA1	IMA0

EIMR bits are set/reset by software

When the IM bit is set (and the global IEN is enabled), an interrupt request is generated if the corresponding IP bit is set. When IM = "0", no request will be generated.

- IMxy = "1": an interrupt request can be acknowledged (depending on IEN)
- IMxy = "0": an interrupt request is masked.

b7 = **IMD1:** *Interrupt Mask of Interrupt Channel D1*

b6 = **IMD0:** *Interrupt Mask of Interrupt Channel D0*

b5 = **IMC1:** *Interrupt Mask of Interrupt Channel C1*

b4 = **IMC0:** *Interrupt Mask of Interrupt Channel C0*

b3 = **IMB1:** *Interrupt Mask of Interrupt Channel B1*

b2 = **IMB0:** *Interrupt Mask of Interrupt Channel B0*

b1 = **IMA1:** *Interrupt Mask of Interrupt Channel A1*

b0 = **IMA0:** *Interrupt Mask of Interrupt Channel A0*

EIPLR R245 (F5h) Page 0 Read/Write
External Interrupt Priority Level Register

Reset value: 1111 1111 (FFh)

7							0
PL2D	PL1D	PL2C	PL1C	PL2B	PL1B	PL2A	PL1A

EIPLR bits are set/reset by software

b7-b6 = **PL1D, PL2D:** *Priority level for the Group INTD0, INTD1*

b5-b4 = **PL1C, PL2C:** *Priority level for the Group INTC0, INTC1*

b3-b2 = **PL1B, PL2B:** *Priority level for the Group INTB0, INTB1*

b1-b0 = **PL1A, PL2A:** *Priority level for the Group INTA0, INTA1*

EIVR R246 (F6h) Page 0 Read/Write
External Interrupt Vector Register

Reset value: xxxx 0110 (X6h)

7							0
V7	V6	V5	V4	TLTEV	TLIS	IAOS	EWEN

b7-b4 = **V7 to V4:** *Most significant nibble of External Interrupt Vector.* Not initialized by reset.

b3 = **TLTEV:** *Top Level Trigger Event bit* When set, the Top Level event is triggered on rising edge of NMI input pin. Triggering on the falling edge of the NMI input pin is activated when this bit is "0" (reset value)

b2 = **TLIS:** *Top Level Input Selection bit* This bit selects the source of the Top Level Interrupt between the external NMI pin (when "1", the reset value) and the Timer/Watchdog End of Count (when "0").

b1 = **IA0S:** *Interrupt A0 Selection bit* When set, the External Interrupt pin is selected as the External Interrupt Channel A0 source. When reset the source is the Timer/Watchdog End of Count interrupt.

b0 = **EWEN:** *External Wait Enable bit.*

Must be held to the reset state.

NICR R247 (F7h) Page 0 Read/Write
Nested Interrupt Control Register

Reset value: 0000 0000 (00h)

7							0
TLNM	HL6	HL5	HL4	HL3	HL2	HL1	HL0

b7 = **TLNM:** *Top Level Not Maskable.*

If TLNM = "1", a top level request is generated as TLIP is set. Once TLNM is set, it can be cleared only with an hardware reset

bx = **HLx:** *Hold Level x* These bits are set to "1" when, in Nested Mode, an interrupt service routine at level x is interrupted from a request with higher priority (other than the Top Level interrupt request). It is cleared by the `iret` execution when the routine at level x is recovered.

NOTES

5 CLOCK

5.1 INTRODUCTION

The ST9 Clock generator module generates the internal clock for the ST9 core and the on-chip peripherals. The Clock generator can be driven by an external crystal circuit, connected to the OSCIN and OSCOUT pins, or by an external pulse generator, connected to OSCIN.

5.2 CLOCK MANAGEMENT

The oscillator circuit generates an internal clock signal with the same period and phase as at the OSCIN input pin. The maximum frequency allowed is 24MHz.

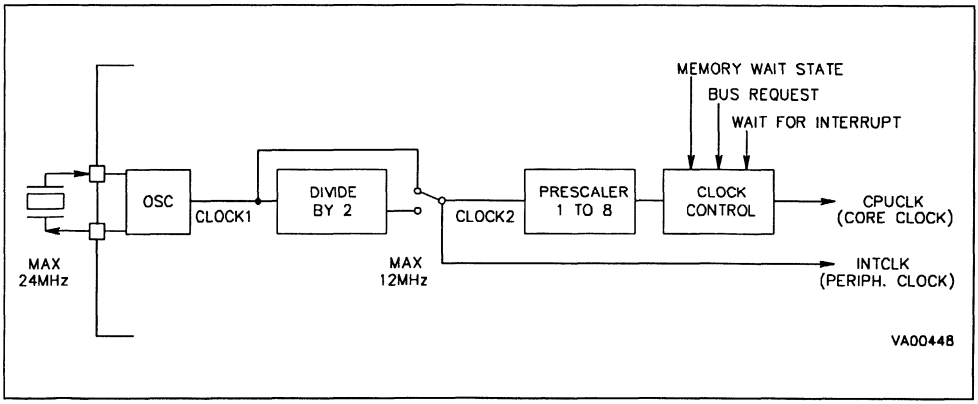
As shown in Figure 5-1, the CLOCK1 signal drives a programmable divider by two. If the control bit MODER.DIV2 (R235.5) is set, the internal clock CLOCK2 is CLOCK1 divided by two; otherwise, if DIV2 bit is cleared, the clock signal CLOCK2 has the same period and phase as CLOCK1. CLOCK2 drives the internal clock INTCLK delivered to all ST9 on-chip peripherals and acts as the central timebase for all timing functions (e.g. Multifunction Timer or Serial Communications Interface Baud Rate generator).
The maximum frequency allowed for INTCLK is 12MHz. For internal operating frequencies above 8MHz, it is recommended to work with the Clock Divider active in order to provide a duty cycle of 50% for INTCLK.

CLOCK2 also drives a programmable prescaler which generates the basic time base, CPUCLK, for the instruction executer of the ST9 core. This allows the user to slow the program execution time to reduce power dissipation, and to locally speed up certain code segments for time critical routines. The internal peripherals are not affected by the CPUCLK prescaler. The prescaler value divides the input clock by the value programmed in the control bits MODER.PRS2,1,0 (R235.4,3,2). If the prescaler value is zero, no prescale is made, thus CPUCLK has the same period and phase as CLOCK2 and INTCLK. If the value is different from 0, the prescaling is equal to the value plus one, ranging thus from two (PRS2,1,0 = 1) to eight (PRS2,1,0 = 7). The clock generated is shown in Figure 5-2. It must be noticed that the prescaling of the clock does not keep the duty cycle to 50%, but stretches the high level of the clock until completion.

When External Memory Wait (or Bus Request or Wait for Interrupt) events occur, CPUCLK is stretched on the high level for the whole period required by the function.

Note. The added wait cycles refer to the INTCLK frequency and not the original CPUCLK.

Figure 5-1. Peripheral and Core Clocks



5.3 CLOCK CONTROL REGISTER

The ST9 clock division by 2 and the clock prescaling are controlled by the MODER register.

Note. This register contains bits with other functions. Only the bits relating to control of the clock are shown here.

MODER R235 (EBh) System Read/Write Mode Register

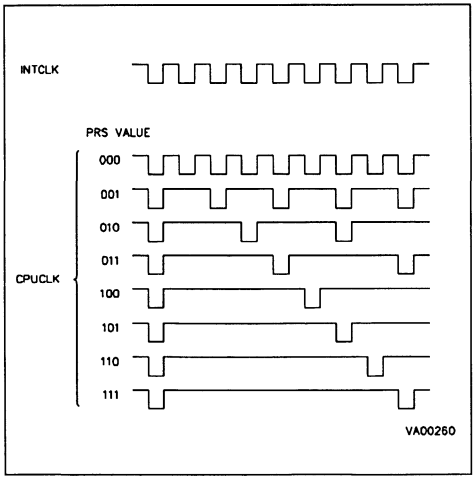
Reset Value : 1110 0000 (E0h)

7							0
X	X	DIV2	PRS2	PRS1	PRS0	X	X

b5 = **DIV2:** *OSCIN Divided by 2.* This bit controls the divide by 2 circuit which operates on the OSCIN Clock. A logical "1" value means that the OSCIN clock is internally divided by 2, and a logical "0" value means that no division of the OSCIN Clock occurs.

b4-b2 = **PRS2, PRS1, PRS0:** *Prescaling of ST9 Clock.* These bits define the prescaler value used to prescale the CPUCLK from INTCLK. When these three bits are reset, the CPUCLK is not prescaled, and is equal to INTCLK; in all other cases, the internal clock is prescaled by the value of (PRS2,1,0 + 1).

Figure 5-2. Core Clock Prescaling



5.4 WAIT CONTROL REGISTER

WCR R252 (FCh) Page 0 Read/Write
WAIT Control Register

Reset Value : 0111 1111 (7Fh)

7							0
X	WDGEN	WDM2	WDM1	WDM0	WPM2	WPM1	WPM0

b7 = Reserved. read as "0"

b6 = **WDGEN:** refer to Timer/WatchDog Chapter

b5-b3 = **WDM2-0:** *Dataspace Wait Cycles.* These bits should be reset to zero after initilization to achieve optimum performance of thecore.

b2-b0 = **WPM2-0:** *Program Space Wait Cycles.* These bits should be reset to zero after initialization to achieve optimum performance of the core.

5.5 OSCILLATOR CHARACTERISTICS

The on-chip oscillator circuit (Figure 5-3) is an inverting gate circuit.

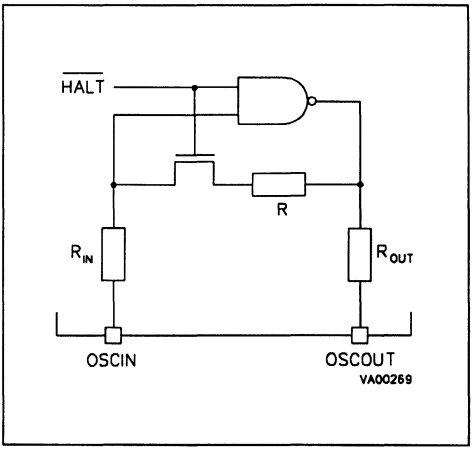
Note. Owing to the Q factor required, Ceramic Resonators may not provide a reliable oscillator source.

In Halt mode, set by means of the HALT instruction, the parallel resistor R is disconnected and the oscillator is disabled, forcing the internal clock CLOCK1 to a high level and OSCOUT to a high level.

To exit the HALT condition and restart the oscillator, an external RESET pulse is required of a minimum duration of 12ms (Figure 5-6).

It must be noted that if the Timer/Watchdog watchdog function is enabled, a HALT instruction will not disable the oscillator. This to avoid stopping the watchdog if, by an error, a HALT code is executed. When this occurs, the ST9 CPU falls into an endless loop ended by the watchdog (or external) reset.

Figure 5-3. Internal Oscillator Schematic



Note: ROUT< 50Ω R > 1MΩ 300Ω < RIN < 500Ω

Figure 5-4. Crystal Oscillator

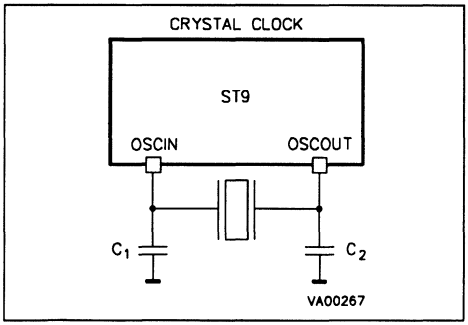


Figure 5-5. External Clock

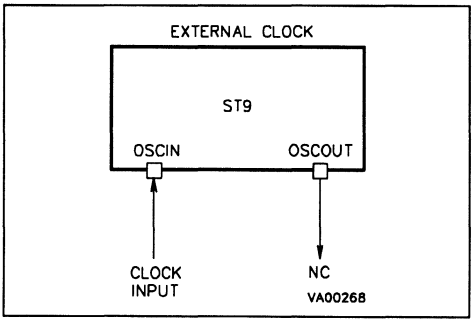


Table 5-1. Crystal Specification (C0 <= 7pF)

Frequency (MHz)	C1=C2=56pF Rs Max	C1=C2=47pF Rs Max	C1=C2=22pF Rs Max
24	20	25	70
16	40	60	150
12	80	100	250
8	180	240	600
4	700	800	600

Table 5-2. Oscillator Transductance

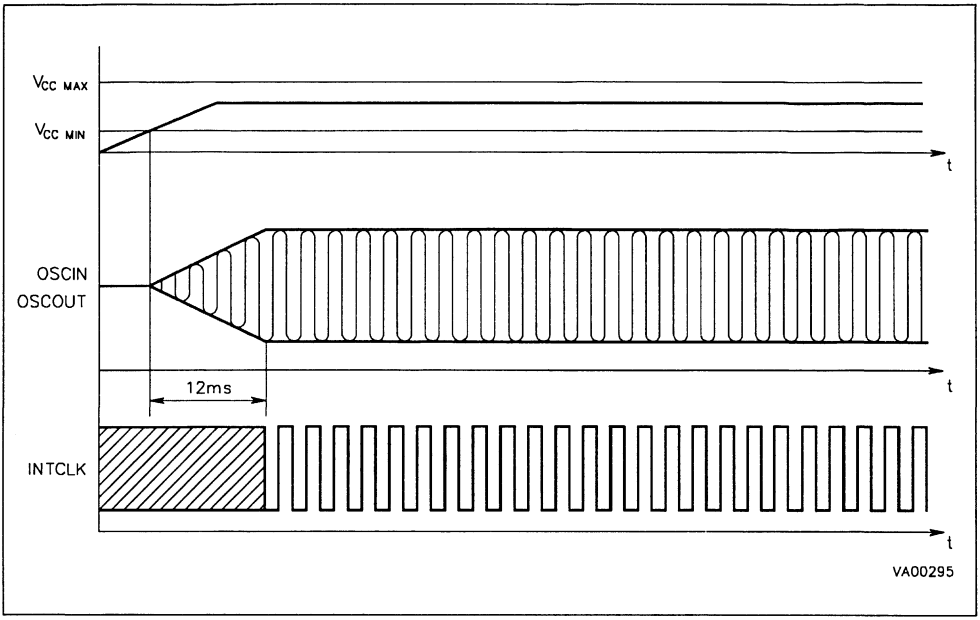
gm	Min	Typ	Max
mA/V	3	5.8	9.5

Legend:
Rs: Parasitic Series Resistance of the quartz crystal (upper limit)
C0: Parasitic capacitance of the quartz crystal (upper limit, < 7pF)
C1, C2: Maximum Total Capacitances on pins OSCIN and OSCOUT(the value includes the external capacitance tied to the pin plus the parasitic capacitance of the board and of the device)
gm: Transconductance of the oscillator

Note. The tables are relative to the fundamental quartz crystal only (not ceramic resonator).

OSCILLATOR CHARACTERISTICS (Continued)

Figure 5-6. Oscillator Start-up Sequence

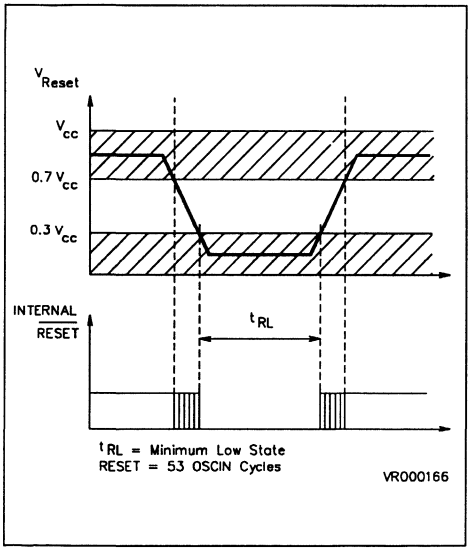


6 RESET

6.1 INTRODUCTION

The processor Reset overrides all the other conditions and forces the ST9 to the reset state. During Reset, the internal registers are set to the reset value, as shown in Table 6-1 for the system and Page 0 Registers and the I/O pins are set to the Bidirectional Weak Pull-up mode (see Warning). The programmer must then initialize the ST9 system and peripheral control registers to give the required functions.

Figure 6-7. Signal to be applied on Reset Pin



6.2 RESET GENERATION

The reset condition can be generated by the external pin $\overline{\text{RESET}}$ or by the on-chip Timer/Watchdog.

The on-chip Timer/Watchdog generates a reset condition if the watchdog mode is enabled (WCR.WDEN cleared, R252 page 0), and if the programmed period elapses without the specific code (AAh,55h) written to the appropriate register. The input pin RESET is not driven low by the on-chip reset generated by the Timer/Watchdog.

During reset, the $\overline{\text{DS}}$ output signal is kept low and the $\overline{\text{AS}}$ output is toggled with the crystal frequency (input at OSCIN) divided by 32. This condition is recognized by off-chip Z-bus peripherals as a reset condition.

6.3 RESET PIN TIMING

The $\overline{\text{RESET}}$ pin has a Schmitt trigger input circuit with hysteresis. The internal reset is generated by the external pin synchronized with the internal clock. The power up reset circuit must keep the $\overline{\text{RESET}}$ input low for a minimum of the crystal startup period plus 53 crystal periods.

Once the $\overline{\text{RESET}}$ pin reaches a logical "1", the processor exits from the reset status after 67 crystal periods from the rising edge ($\overline{\text{DS}}$ is set). The processor then fetches from Program Memory locations 0 and 1 (power-on reset vector) and begins program execution from the address contained in the vector. If the ST9 is a ROMLESS version, without on-chip program memory, ports Port0, Port1 (and Port6 for ST909x family) are set to external memory mode (i.e Alternate Function) and the memory accesses are made to external Program memory with wait cycles insertion.

WARNING: *I/O pins are set to the Weak Pull-up mode during the Reset cycle. This state is forced during the reset sequence, but the I/O pins can be in a random state for up to 64 crystal periods.*
The application circuit must take this into account if it can lead to critical situations in the external circuitry.

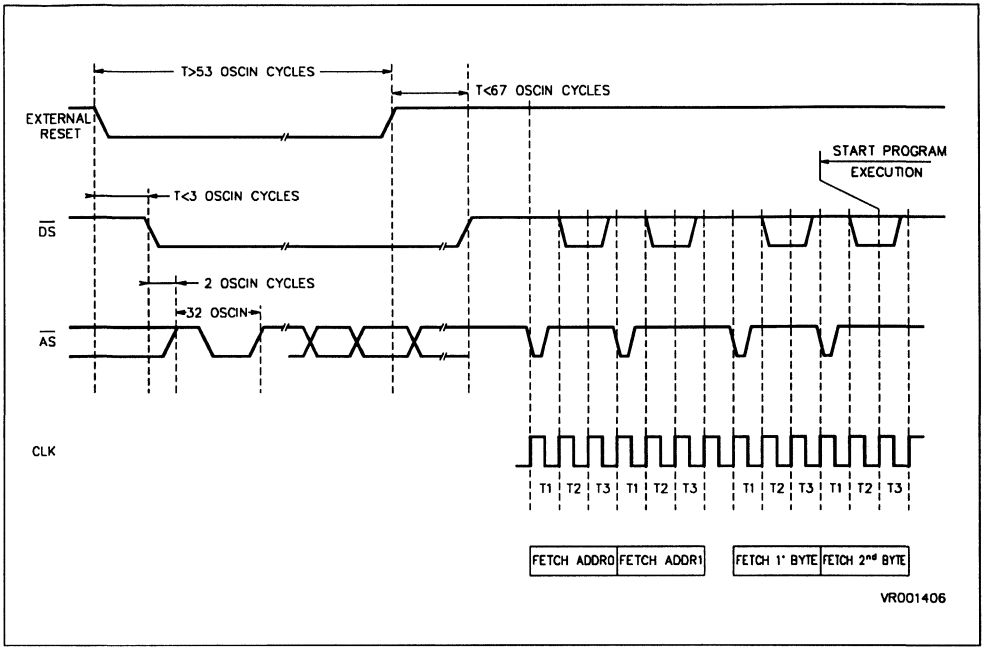
RESET (Continued)

Table 6-1. Internal Registers Reset Values

Register Number	System Register	Reset Value	Page 0 Register	Reset Value
F	(SSPLR)	undefined	Reserved	
E	(SSPHR)	undefined	(SPICR)	00h
D	(USPLR)	undefined	(SPIDR)	undefined
C	(USPHR)	undefined	(WCR)	7Fh
B	(MODER)	E0h	(WDTCR)	12h
A	(Page Ptr)	undefined	(WDTPR)	undefined
9	(Reg Ptr 1)	undefined	(WDTLR)	undefined
8	(Reg Ptr 0)	undefined	(WDTHR)	undefined
7	(FLAGR)	undefined	(NICR)	00h
6	(CICR)	87h	(EIVR)	x2h
5	(PORT5)	FFh	(EIPLR)	FFh
4	(PORT4)	FFh	(EIMR)	00h
3	(PORT3)	FFh	(EIPR)	00h
2	(PORT2)	FFh	(EITR)	00h
1	(PORT1)	FFh	Reserved	undefined
0	(PORT0)	FFh	Reserved	

RESET (Continued)

Figure 6-8. Exit From Reset Timing



NOTES

7 I/O PORTS

7.1 INTRODUCTION

The ST9 is provided with dedicated lines for input/output. These lines, grouped into 8-bit ports, can be independently programmed to provide parallel input/output, or to carry input/output signals to/from the on-chip peripherals and Core (e.g. SPI and OSD). All ports have active pull-ups and pull-down resistors compatible with TTL loads. In addition, pull-ups can be turned off for open-drain operation and weak pull-ups can be turned on to save off-chip resistive pull-ups. Input buffers can be either TTL or CMOS compatible.

7.2 CONTROL REGISTERS

Each port PX has a Data Register PX, and three associated control registers (PXC0, PXC1, PXC2) which define the port line configuration and allow dynamic change in port configuration during program execution. Ports and control registers are mapped into the Register File as shown in Figure 7-1 Ports and control registers are treated like any other general-purpose register. There are no special instruction for port manipulation, any instruction that addresses a register can address the ports. Data can be directly accessed in the port register, without passing through other memory or "accumulator" locations.

Figure 7-1. I/O Register Maps

Applicable to ST9293

GROUP E			GROUP F		
			PAGE 2	PAGE3	
		FFh	Reserved	Reserved	
		FEh	P3C2	Reserved	
		FDh	P3C1	Reserved	
		FCh	P3C0	Reserved	
		FBh	Reserved	Reserved	
		FAh	P2C2	Reserved	
		F9h	P2C1	Reserved	
		F8h	P2C0	Reserved	
		F7h	Reserved	Reserved	
		F6h	P1C2	P5C2	
E5h	P5DR	R229 F5h	P1C1	P5C1	
E4h	P4DR	R228 F4h	P1C0	P5C0	
E3h	P3DR	R227 F3h	Reserved	Reserved	
E2h	P2DR	R226 F2h	P0C2	P4C2	
E1h	P1DR	R225 F1h	P0C1	P4C1	
E0h	P0DR	R224 F0h	P0C0	P4C0	

CONTROL REGISTERS (Continued)

During the reset state, all the Ports are set as bidirectional/weak pull-up mode except for P3.3 and P3.4 which have no weak pull-up, with the output data register set to FFh. This condition is also held after reset (see Warning on page 52) and can be redefined under software control at any time.

7.3 PORT BIT STRUCTURE AND PROGRAMMING

By programming the control bits PXC0.n and PXC1.n (see Figure 7-2) it is possible to configure bit PX.n as Input, Output, Bidirectional or Alternate Function Output, where X is the number of the I/O port, and n the bit within the port (n = 0 to 7).

When programmed as input, it is possible to select the input level as TTL or CMOS by programming the control bit PXC2.n.

The output buffer can be programmed as Push-pull or Open-drain. A Weak Pull-up configuration can be used when the port bit is programmed as Bidirectional. This is an Open-drain configuration with a high pull-up resistor value (turned on by writing a "1"), to avoid the requirement for external resistances.

The basic structure of the bit PX.n of a general purpose port PX is shown in Figure 7-3.

Independently to the chosen configuration, when the User addresses the port as an destination register of an instruction, the port is written to and the data is transferred from the internal Data Bus into the Output Master Latches. When the port is addressed as a source register for an instruction, the port is read and the data stored in the Input Latch is transferred onto the internal Data Bus.

When PX.n is programmed as Input: (Figure 7-4)

- The Output Buffer is forced tristate
- The data present on the I/O pin is sampled into the Input Latch at the beginning of the execution of the instruction which is accessing the port.
- The data stored in the Output Master Latch is copied into the Output Slave Latch at the end of the execution of each instruction. So if bit PX.n is reconfigured as Output or Bidirectional, the data stored in the Output Slave Latch is reflected on the I/O pin.

Figure 7-2. Control Bits

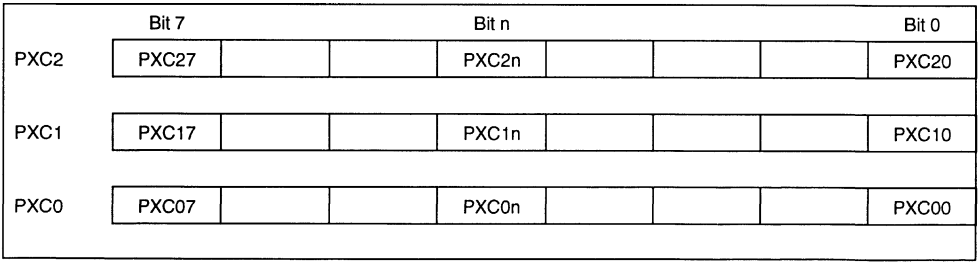


Table 7-1. Port Bit Configuration Table

PXC2n PXC1n PXC0n	0 0 0	1 0 0	0 1 0	1 1 0	0 0 1	1 0 1	0 1 1	1 1 1	
PXn Configuration	BID	BID	OUT	OUT	IN	IN	AF	AF	
PXn Output	WP	OD	PP	OD	HI	HI	PP	OD	HI(1)
PXn Input	TTL	TTL	TTL	TTL	CMOS	TTL	TTL	TTL	0(1)

Legend:
X = Port
n = Bit
AF = Alternate Function
HI = High Impedance
OD = Open Drain
WP = Weak Pull-up
PP = Push-Pull
BID = Bidirectional
TTL = TTL Standard Input
CMOS = CMOS Standard Input
IN = Input
OUT = Output

Note 1: For A/D Converter inputs

PORT BIT STRUCTURE AND PROGRAMMING (Continued)

Figure 7-3. Basic Structure of an I/O Port Pin

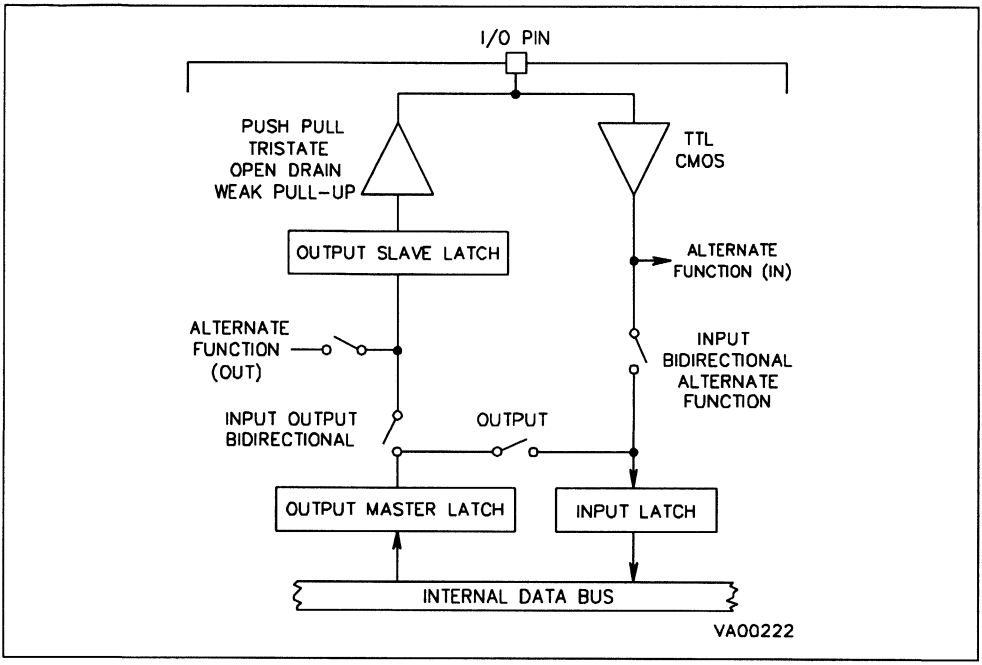


Figure 7-4. Input Configuration

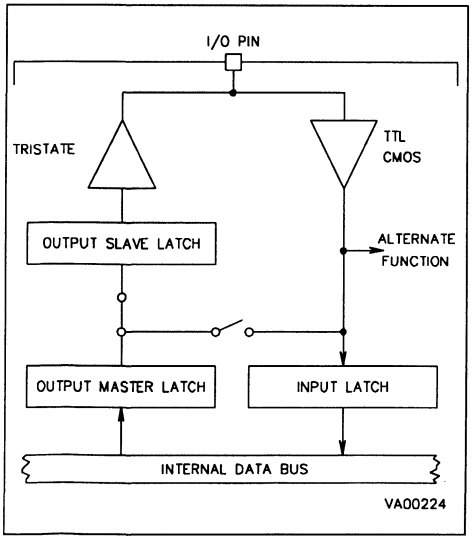
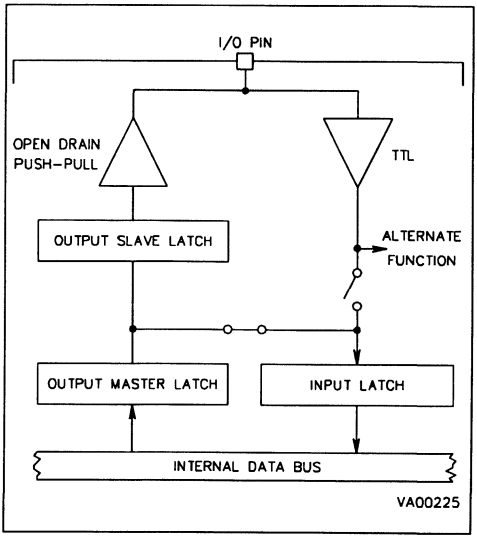


Figure 7-5. Output Configuration



PORT BIT STRUCTURE AND PROGRAMMING (Continued)

When PX.n is programmed as Output: (Figure 7.5)

- The Output Buffer is turned on in an Open-drain or Push-pull configuration
- The data stored in the Output Master Latch is copied both into the Input Latch and into the Output Slave Latch, driving the I/O pin, at the end of the execution of each instruction.

When PX.n is programmed as Bidirectional: (Figure 7-6)

- The Output Buffer is turned on in an Open-drain or Weak Pull-up configuration
- The data present on the I/O pin is sampled into the Input Latch at the beginning of the execution of each instruction
- The data stored in the Output Master Latch is copied into the Output Slave Latch, driving the I/O pin, at the end of the execution of each instruction.

WARNING. Due to the unique feature of the bidirectional mode of reading the external pin instead of the output latch, particular care must be taken with arithmetic/logic and boolean instructions performed on a bidirectional port pin.

These instructions use a read-modify-write sequence, and the result written in the port register depends on the logical level present on the external pin.

This may bring unwanted modifications to the port output register content.

For example:

Port register content	external port value
0Fh	03h

(Bits 3 and 2 are externally forced to 0)

Making a `bset` instruction on bit 7 will return:

Port register content	external port value
83h	83h

(Bits 3 and 2 have been cleared.)

To avoid this situation, it is suggested that all the operations on a port, using at least one bit in bidirectional mode, are performed on a copy of the port register, then transferring the result with a load instruction to the I/O port.

When PX.n is programmed as Alternate Function Output (Figure 7-7) except for Analog Inputs :

- The Output Buffer is turned on in an Open-drain or Push-pull configuration
- The data present on the I/O pin is sampled into the Input Latch at the beginning of the execution of each instruction
- A signal coming from an on-chip Function is allowed to load the Output Slave Latch driving the I/O pin. Signal timing is under control of the Function. If no Function is connected to PX.n the I/O pin is driven to a high level in Push-pull configuration and is driven to high impedance in open drain configuration.

Figure 7-6. Bidirectional Configuration

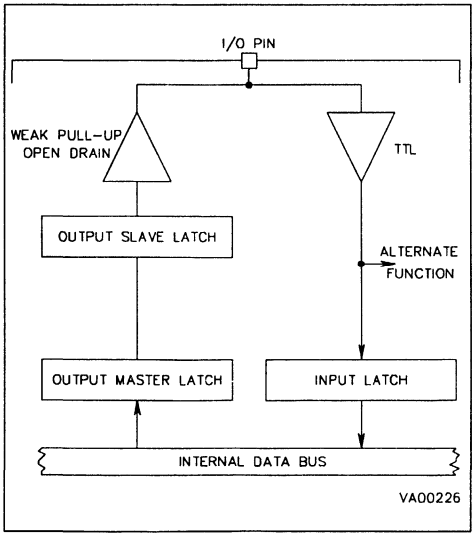
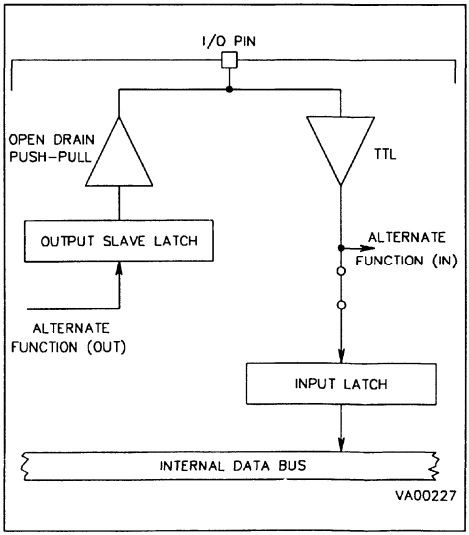


Figure 7-7. Alternate Function Configuration



7.4 ALTERNATE FUNCTION ARCHITECTURE

Each single I/O pin may access three different types of ST9 internal signals:

- Data bus line (I/O)
- 'Alternate Function' Input
- Alternate Function Output

Each pin configuration is made by software, thus allowing the User to choose the type of signal to access a pin. The choice of type of signal is made with the registers PXC2, PXC1, PXC0 of the I/O Port X (Please refer to the previous section for more details)

Pins Declared as an I/O
A pin declared as an I/O is a pin connected to the I/O buffer. In such a case, this pin may either be an Input or an Output or an I/O depending on the value stored in (PXC2, PXC1, PXC0)

Figure 7-8. Example of 3 Alternate Function Inputs

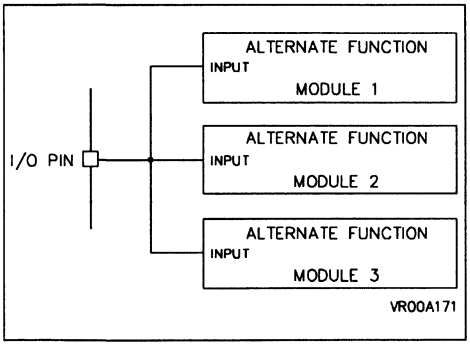
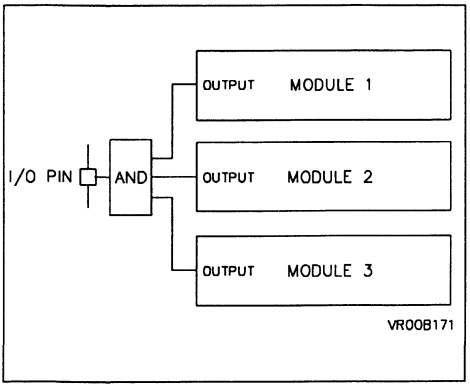


Figure 7-9. Example of 3 Alternate Function Inputs



Pin Declared As An 'Alternate Function' Input
A single pin may be directly connected to several Alternate Function inputs. In such a case, the User has to select the required input mode (TTL or CMOS levels) and to enable, by software, the selected Alternate Function module (by enabling it) and unselect all other Alternate Functions (by disabling them).
No specific configuration of the port is required to enable the input Alternate Function, as the input buffer is directly connected to each module using it. As more than one module can use the same input Alternate Function line, it is under User software control to enable a module to use the input Alternate Function.
The digital I/O remains operational even when using the Alternate Function input. The exception to this is for an I/O port bit connected to analog voltages (for the Analog to Digital Converter, see Figure 9.10).

Pin Declared As An Alternate Function Output
A pin declared as an Alternate function output corresponds to (PXC2,PXC1,PXC0) = 1,1,1 or 0,1,1. Several Alternate Function outputs may drive a common pin. In such a case, the Alternate Function output signals are ANDed before driving the common pin. The User has therefore to select, by software, the Alternate Function Output required by enabling it and disabling all other Alternate Function Outputs on the same pin (a disabled Alternate Function Output outputs a "1").

The inputs to on-chip Functions and Alternate Function Outputs are predefined for each I/O pin. Please refer to the Alternate Function Table at the beginning of this datasheet for the exact configuration.

WARNING :

The user must take care not to program I/O pins of 8-bit ports which are not externally available in the 42 pin package to the input tri-state mode. This is to avoid spurious and extra power consumption that may appear due to CMOS floating inputs.

ALTERNATE FUNCTION (Continued)

General Configuration
A single pin may be used, according to different phases of the software, as an I/O or connected to an input or an Alternate Function output.

WARNING: *When a pin is connected to an Input Function and to an Alternate Function output, the User must be aware of the fact that the Alternate Function output signal always input to the Alternate Function module(s) declared as input(s).*

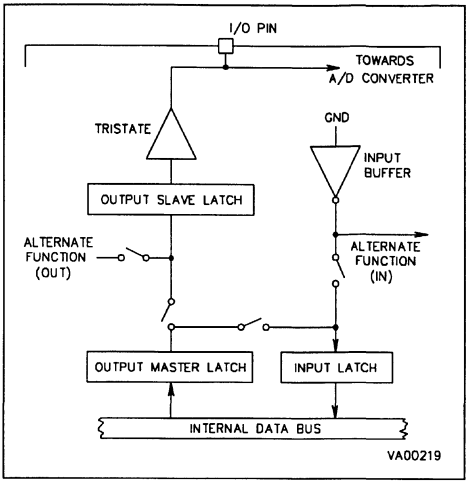
7.5 SPECIAL PORTS

7.5.1 Bit Structure for A/D Converter Inputs

When a port bit is used as input for an on-ship A/D Converter, its structure is modified as shown in Figure

The behaviour of this bit is identical to the general purpose bit described in paragraph 9.68 except when it is programmed as Alternate Function. In this case, the Output Buffer is forced Tristate and the input of the Input Buffer is disconnected from the I/O pin and forced low. In this way the I/O pin is free to assume any analog value without causing power consumption in the Input Buffer. The bit **MUST** be programmed to **(PXC2, PXC1, PXC0)=1,1,1)** to assume this special configuration.

Figure 7-10. A/D Input Port Bit Structure



7.6 I/O STATUS AFTER WFI, HALT AND RESET

The status of the ST9 I/O ports during the Wait For Interrupt, Halt and Reset operational modes is set to the bidirectional weak pull-up state, except for P3.3 and P3.4 which have no weak pull-up.

WARNING : *I/O pins are set to the Weak Pull-up mode during the Reset cycle. This state is forced during the reset sequence, but the I/O pins can be in a random state for up to 64 crystal periods.*
The application circuit must take this into account if it can lead to critical situations in the external circuitry.

8 SERIAL PERIPHERAL INTERFACE

8.1 INTRODUCTION

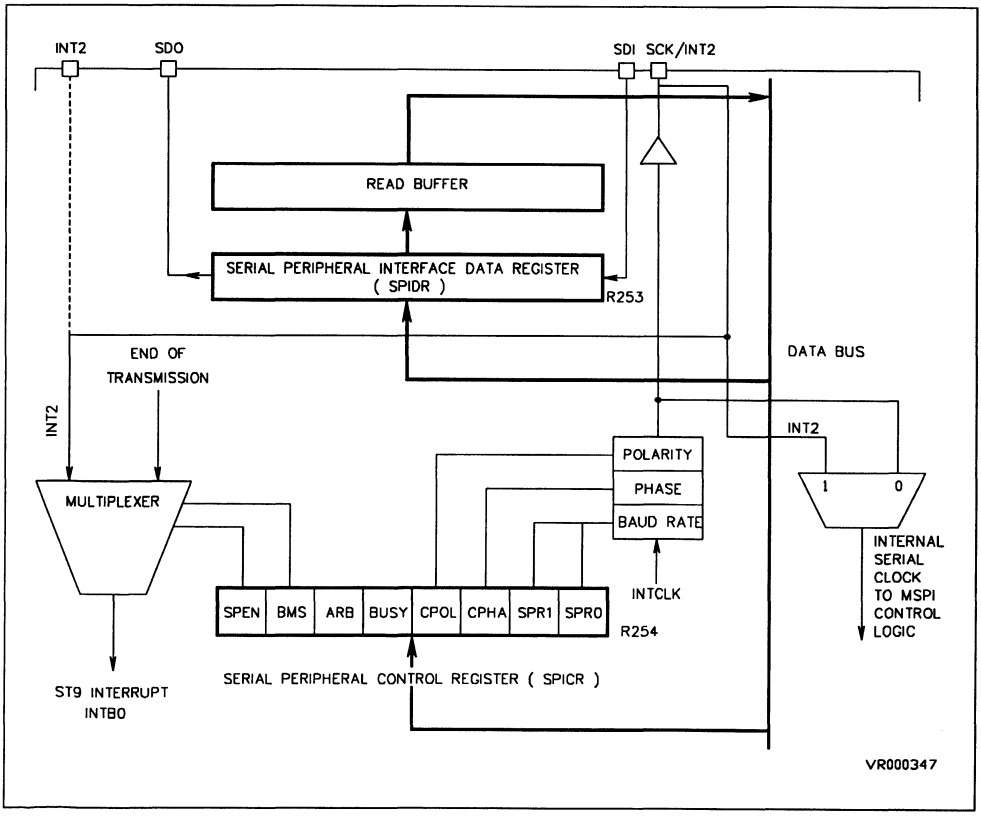
The Serial Peripheral Interface (SPI) is integrated into the Core module of the ST9 and provides a general purpose shift register based peripheral allowing several external peripherals to be linked through an SPI protocol bus. In addition, special modes allow reduced software overhead with I^2C-bus and IM-bus Communication standards.

The SPI uses 3 lines comprising Serial Data In (SDI) and Alternate Function outputs Serial Data Out (SDO) and Synchronous Serial Clock (SCK). Additional I/O pins may act as device selects or IM-bus address ident signals.

Its Main Features are:

- Full duplex 3-wire synchronous transfer
- Master operation only
- 1.5MHz max bit transfer frequency (INTCLK = 12MHz)
- 4 Programmable bit rates
- Programmable clock polarity and phase
- Busy Flag
- End of transmission interrupt
- Additional hardware to facilitate more complex protocols

Figure 8-1. Block Diagram



8.2 FUNCTIONAL DESCRIPTION

The SPI, when enabled, receives input data from the ST9 Core internal data bus into SPIDR, and originates the Serial Clock (SCK) based upon dividing of the internal processor clock (INTCLK). The data is parallel loaded into the 8 bit shift register (from the internal bus) during a write cycle and then shifted out serially through the SDO pin (Most Significant bit first) to the slave device, which responds by sending its data to the master device via the SDI pin. This implies full duplex transmission with data-out and data-in both synchronized with the same clock signal. Thus the transmitted byte is replaced by the byte received, eliminating the need to have separate "Tx empty" and "Rx full" status bits.

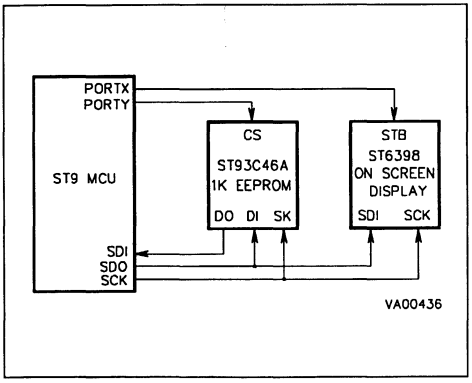
When the shift register is loaded, data is parallel transferred to the read buffer and data becomes available for the ST9 during a following read cycle.

The SPI requires three pins on an I/O port:

SCK	Serial Clock signal
SDO	Serial Data Out
SDI	Serial Data In

An additional output bit of an I/O port may be used to perform the slave chip select signal.

Figure 8-2. A Typical SPI Network



8.2.1 Input Signal Description

Serial Data In (SDI)

Data is transferred serially from a slave to a master on this line, most significant bit first. In an S-BUS/I^2C-bus configuration, SDI line senses the value forced on the data line (by SDO or by another peripheral connected to the S-bus/I^2C-bus environment).

8.2.2 Output Signal Description

Serial Data Out (SDO)

The SDO pin is configured as an output for the master device. This is obtained by programming the corresponding I/O pin as an output alternate function. Data is transferred serially from a master to a slave on SDO, most significant bit first. This pin is forced to the high impedance state when the SPI is disabled and is set to "1" when arbitration is lost (during an S-bus/I^2C-bus protocol transmission). The master device always allows data to be applied on the SDO line one half cycle before the clock edge in order to latch the data for the slave device.

Master Serial Clock (SCK)

The master device uses SCK to latch the incoming data on the SDI line. This pin is forced to a high impedance state when SPI is disabled (SPEN, SPICR.7 = "0"), in order to avoid clock contention from different masters in a multi-master system.
The master device generates SCK from INTCLK. SCK is used to synchronize the transfer of data both in and out of the device through its SDI and SDO pins. The SCK type and its relationship to data are controlled by the CPOL and CPHA bits in the Serial Peripheral Control Register.
This input is provided with a digital filter which cleans spikes lasting less than one INTCLK period.

Two bits (SPR1 and SPR0) in the Serial Peripheral Control Register, SPICR (R254) select the clock rate. Four frequencies can be selected, two in a high frequency range (mostly used with the SPI protocol) and two in a medium frequency range (mostly used for more complex protocols).

8.3 INTERRUPT STRUCTURE

SPI peripheral is associated with external interrupt channel B0 (pin INT2). Multiplexing between the external pin and SPI internal source is controlled by the SPEN and BMS bits according to the following table.

The two possible SPI interrupt sources are: End of transmission (after each byte) and S-bus/I²C-bus start condition. Care should be taken when toggling SPEN or/and BMS bits from (0,0) status, this should be done by masking the interrupt channel B0 (reset of EIMR.IMB0, bit 2 of External Interrupt Mask Register). Furthermore it is necessary to clear possible spurious requests on the corresponding channel by resetting the interrupt pending bit EIPR.IPB0 (bit 2 of External Interrupts Pending Register).

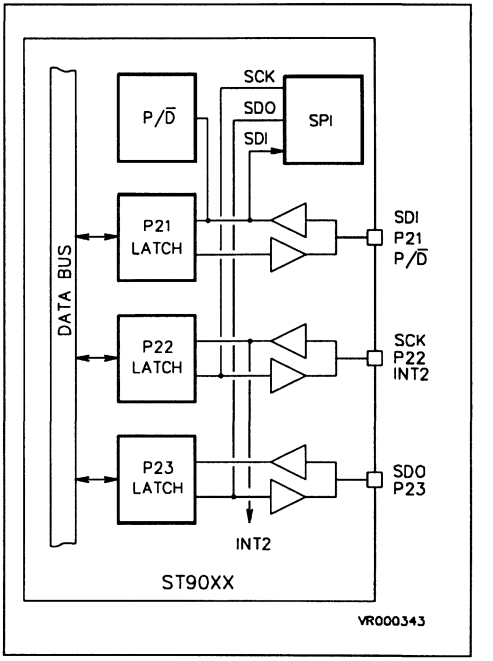
The INT2 input Function is always mapped together with the SCK input Function to allow start/stop bit detection when using S-bus/I²C-bus protocols.

A delay instruction (e.g. a `NOP` instruction) should be inserted between the SPEN toggle instruction and the interrupt pending bit reset instruction.

Table 8-1. Interrupt Configuration

SPEN	BMS	Interrupt Source
0	0	External channel INT2
0	1	S-bus/I²C bus start or stop condition
1	X	End of one byte transmission

Figure 8-3. SPI I/O Pins



8.4 SPI REGISTERS

SPI uses two registers mapped on page 0 of the register file:

SPIDR R253 (FDh) Page 0 Read/Write
SPI Data Register

Reset Value: 0000 0000b (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

b7-b0 = **D0-D7:** *SPI Data Bits.* This register contains the data transmitted and received by the SPI. Data is transmitted b7 first, and receives incoming data into b0. Transmission is started by writing to this register.

SPICR R254 (FEh) Page 0 Read/Write
SPI Control Register

Reset Value: 0000 0000b (00h)

7							0
SPEN	BMS	ARB	BUSY	CPOL	CPHA	SPR1	SPR0

b7 = **SPEN:** *Serial Peripheral Enable.* When set, the two alternate functions SCK and SDO are enabled. When disabled, SCK and SDO are kept in high impedance. Furthermore, SPEN affects the selection of the source for interrupt channel B0. Transmission will start by simply writing the data into the SPIDR Register.

b6 = **BMS:** *S-bus/I²C-bus Mode Selector.* This bit should be set to "1" when the SPI is used in an S-bus/I²C-bus protocol. It enables S-bus/I²C-bus arbitration, clock synchronization and Start/ Stop detection.
When this bit is reset to "0", a reinitialisation of the SPI logic is performed allowing recovery procedures after a R_X/T_X failure. BMS (and SPEN) affects the selection of the source for interrupt channel B0.

b5 = **ARB:** *Arbitration flag bit.* This bit is set when the SPI, in S-bus/I²C-bus mode, loses arbitration, and is reset when an S-bus/I²C-bus stop condition is detected. ARB can be reset by software. When ARB is set automatically, the SDO pin is set to high value until a write instruction on SPIDR is performed.

b4 = **BUSY:** *SPI Busy Flag.* BUSY flag is set when a transmission is in process. This bit allows the user to monitor the SPI status by polling its value.

b3 = **CPOL:** *Transmission Clock Polarity.* CPOL controls the normal or steady state value of the clock when data is not being transferred.

As the SCK line is held in a high impedance state when the SPI is disabled (SPEN = "0"), the SCK pin must be connected to V_{SS} or V_{CC} through a resistor according to the CPOL state. Polarity should be selected during the reset routine according to the value set into all peripherals and must not be changed during program execution.

b2 = **CPHA:** *Transmission Clock Phase.* CPHA controls the relationship between the data on the SDI and SDO pins and the clock produced at the SCK pin. CPHA bit selects the clock edge which captures data and allows it to change state. It has its greatest impact on the first bit transmitted (MSB) because it does (or does not) allow a clock transition before the first data capture edge.
Figure 8-5 shows the relationship between CPHA, CPOL and SCK, and indicates active clock edges and strobe times.

CPOL	CPHA	SCK on Figure 8-5
0	0	(a)
0	1	(b)
1	0	(c)
1	1	(d)

b1-b0 = **SPR1,SPR0:** *SPI Rate.* These two bits select one (out of four) baud rates to be used as SCK.

SPR1	SPR0	Clock Divider	SCK Frequency (INTCLK = 12MHz)	
0	0	8	1500kHz	(T = 0.67µs)
0	1	16	750kHz	(T = 1.33µs)
1	0	128	93.75kHz	(T = 10.66µs)
1	1	256	46.87kHz	(T = 21.33µs)

The data on the SDA line is sampled with the low to high transition on the SCL line.

8.5 WORKING with DIFFERENT PROTOCOLS

The SPI peripheral offers the following facilities to work with S-bus/I²C-bus and IM-bus protocols:

- Interrupt request on start/stop detection
- Hardware clock synchronisation
- Arbitration lost flag with an automatic set of data line

Note that the I/O bit associated to the SPI should be returned to a defined state as a normal I/O pin before changing the SPI protocol.

The following paragraphs provide information to manage these protocols.

8.5.1 I²C-bus Interface

I²C-bus is a two-wire bidirectional data-bus, the two lines being SDA (Serial DAta) and SCL (Serial CLock). Both are open drain lines to allow arbitration. As shown in figure 8-6, data is toggled with clock low and Start and Stop conditions are detected when a high to low (start) or a low to high (stop) transition on the SDA line occurs with the SCL line high.

Each transmission consists of nine clock pulses (SCL line). The first 8 pulses transmit the byte (msb first), the ninth is used by the receiver to acknowledge.

Figure 8-4. S-Bus/I²C-bus Peripheral Compatibility without S-Bus Chip Select

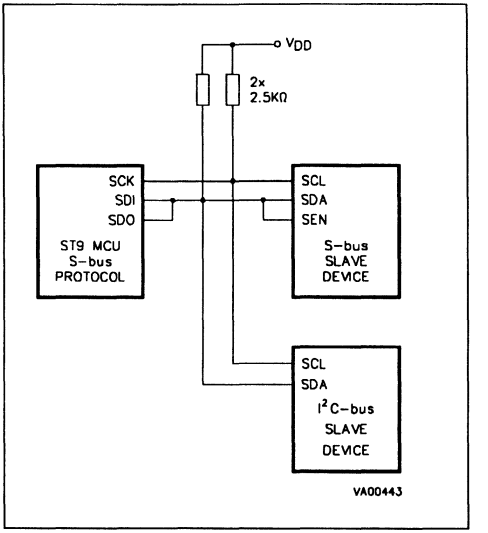
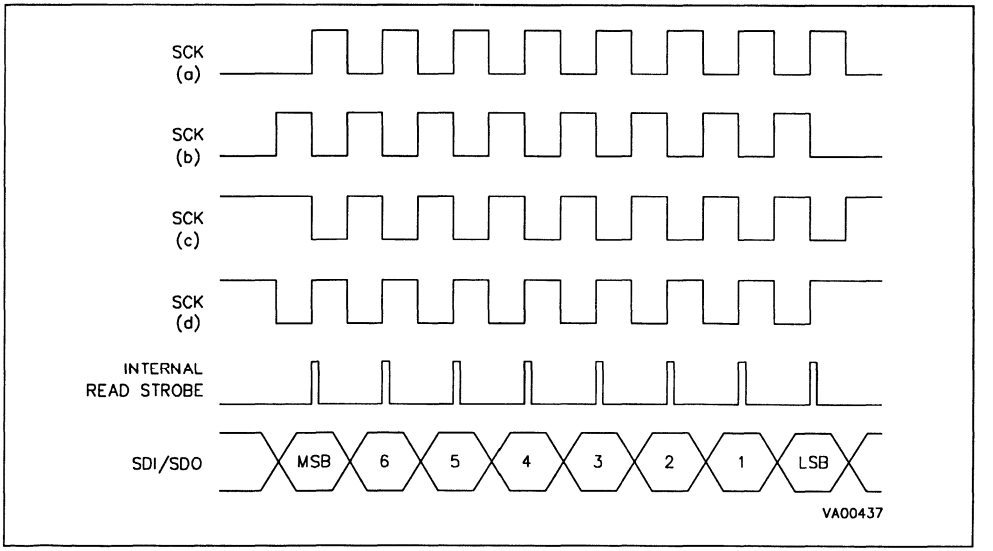


Figure 8-5. SPI Data and Clock Timing

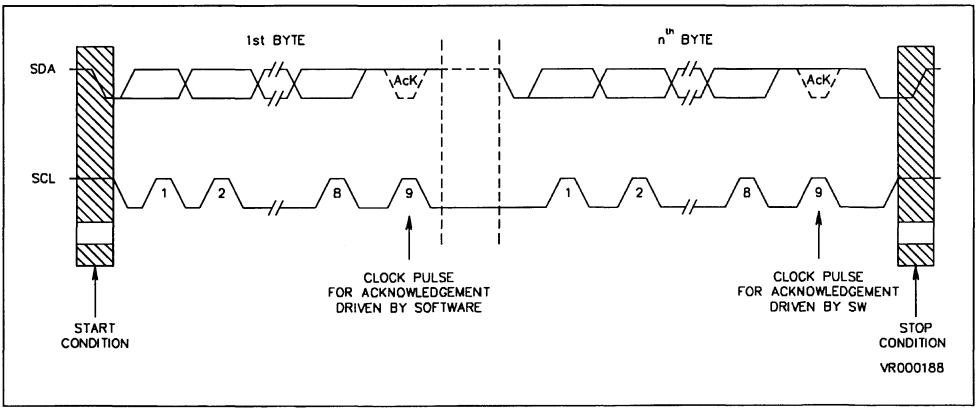


DIFFERENT PROTOCOLS (Continued)

Table 8-2. Typical I^2C-bus Sequences

Phase	Software	Hardware	Notes
INITIALIZE	SPICR.CPOL, CPHA = 0, 0 SPICR.SPEN = 0 SPICR.BMS = 1 SCK pin set as AF output SDI pin set as input Set SDO port bit to 1	SCK, SDO IN HI-Z SCL, SDA = 1, 1	Set polarity and phase SPI disable START/STOP interrupt Enable
START	SDO pin set as output Open Drain Set SDO port bit to 0	SDA = 0, SCL = 1 interrupt request	START condition receiver START detection
TRANSMISSION	SPICR.SPEN = 1 SDO pin as Alternate Function output load data into SPIDR	SCL = 0 Start transmission interrupt request	Managed by interrupt routine load FFh when receiving end of transmission detection
ACKNOWLEDGE	SPICR.SPEN = 0 Poll SDA line Set SDA line SPICR.SPEN = 1	SCK, SDO in HI-Z SCL, SDA = 1 SCL = 0	SPI disable only if transmitting only if receiving only if transmitting
STOP	SDO pin set as output Open Drain SPICR.SPEN = 0 Set SDO port bit to 1	SDA = 1 interrupt request	STOP condition

Figure 8-6. SPI Data and Clock Timing



DIFFERENT PROTOCOLS (Continued)

SPI Working With I²C-bus

To use the SPI with the I²C-bus protocol, the SCK line is used as SCL, the SDI and SDO lines, externally wired-OR'd, are used as SDA. All the output pins must be configured as open drain (see Figure 8-6).

When programmed to the I²C-bus protocol, the SPI functions up to a data baud rate of 750kHz (SPR1=0, SPR0=1) for fosc=12MHz

Table 8-2 shows the typical I²C-bus sequence divided in 5 phases: initialize, start, transmission, acknowledge and stop. Software and hardware will take care of each phase. According to this example, a master to slave transmission can be managed.

During the transmission phase, the following I²C-bus features are also supported by hardware.

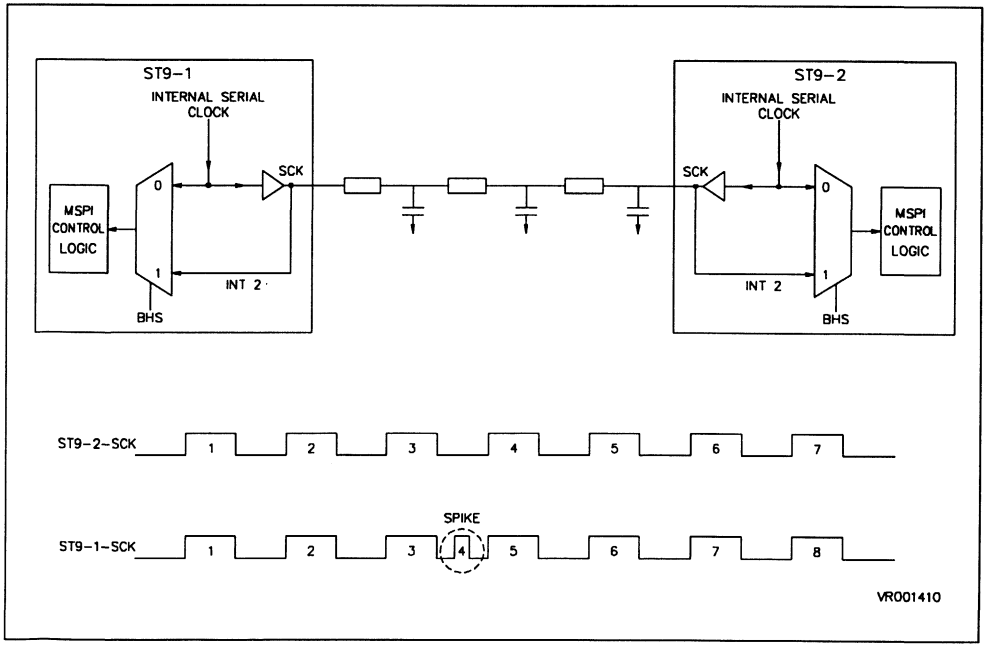
Clock Synchronization

In a multimaster I²C-bus system, when more masters generate their own clock, synchronization is needed. The first master which releases the SCL line stops internal counting, restarting only when the SCL line goes high (released by all the other masters). In this way, devices using different clock sources and different frequencies can be interfaced.

Arbitration Lost

When more masters are sending data on SDA line, the following mechanism is performed: if the transmitter sends a "1" and SDA line is forced low by another device the ARB flag (SPICR.5) is set and the SDO buffer is "switched off. (ARB is reset and SDO buffer is "switched on" when SPIDR is written to again). When BMS is set to "1" the peripheral clock is supplied through the INT2 line by the external clock line (SCL). Due to potential noise spikes (which must last longer than one INTCLK period to be detected), RX or TX may gain a clock pulse.
Referring to Figure 8-7, if ST9-1 detects a noise spike and gains a clock pulse, it will stop its transmission in advance and hold the clock line low causing ST9-2 to be frozen at the 7th bit. To exit and recover from this condition the BMS bit must be reset to "0", this will cause the reset of the SPI logic, aborting the current transmission. An End of Transmission interrupt is generated after this reset sequence.

Figure 8-7. SPI Arbitration



DIFFERENT PROTOCOLS (Continued)

8.5.2 S-Bus Interface

S-bus is a three-wire bidirectional data-bus, with functional features similar to I^2C-bus. Differently from I^2C-bus, the START/STOP conditions are given by encoding the information on 3 wires instead of 2, as shown in Figure 8-8. The additional line is referred as SEN.

SPI Working With S-bus

The S-bus protocol uses the same pin configuration as I^2C-bus for generating the SCL and SDA lines. The additional SEN line is managed through a standard ST9 I/O port under software control (see Figure 8-9).

Figure 8-8. Mixed S-bus and I^2C-bus system

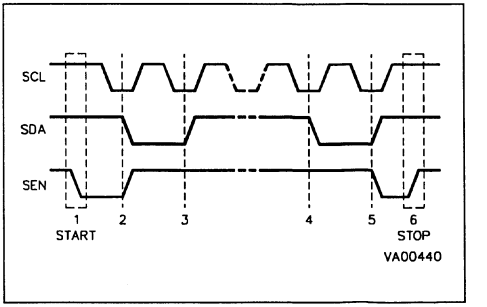


Figure 8-9. S-bus Configuration

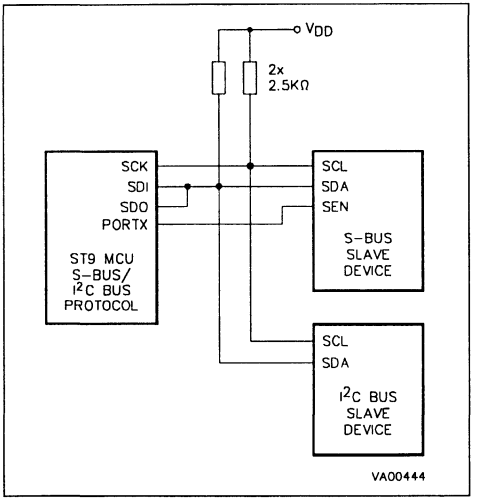
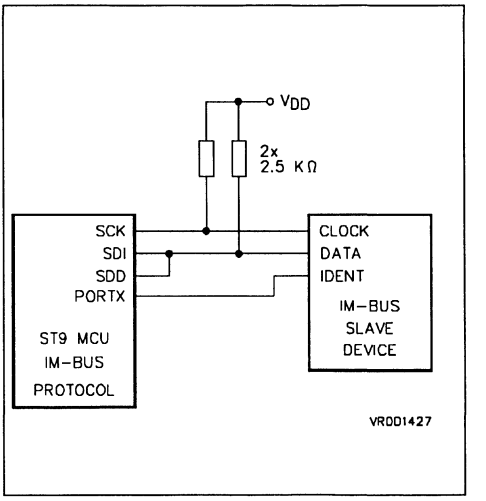


Figure 8-10. ST9 and InterMetal Peripheral



DIFFERENT PROTOCOLS (Continued)

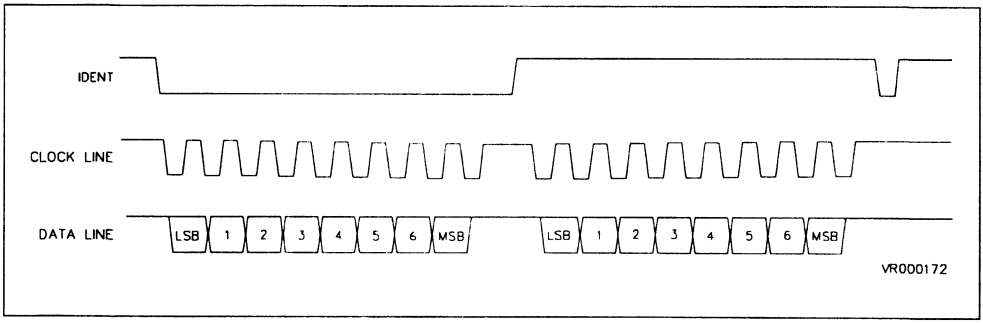
8.5.3 IM-Bus Interface

The IM-bus has a bidirectional data line and a clock line, and in addition it requires an IDENT line that distinguishes an address from a data byte (Figure 8-11). Unlike the I^2C-bus protocol, the IM-bus protocol sends the least significant bit first, this requires a software routine which reverses the bit order before sending, and after receiving a data byte Figure 8-10 shows the connections for an IM-bus peripheral to an ST9 SPI. The SDO and SDI pins are connected to the bidirectional data pin of the peripheral device. The SDO alternate function is set in Open Drain (external 2.5KΩ pull-up resistors are required).

With this type of configuration, data is sent to the peripheral by writing the data byte to SPIDR. To receive data from the peripheral, the User should write FFh into SPIDR in order to generate the shift clock pulses. As the SDO line is set to the Open Drain configuration, the incoming data bits that are set to one do not affect the SDO/SDI line status (which defaults to a high level due to the FFh in the transmit register), while incoming bits that are set to "0" pull the input line low.

In software it is necessary to initialise the ST9 SPI with CPOL and CPHA set to "1", "1". By using a general purpose I/O as the IDENT line and forcing it to a logical "0" when writing to SPIDR, an address is sent (or read). Then, by setting this bit to a logical "1" and writing to SPIDR, data is sent to the peripheral. When all the address and data pairs are sent it is necessary to drive the IDENT line low and high to create a short pulse. In this way the stop condition is generated.

Figure 8-11. IM bus Timing



NOTES

9 TIMER/WATCHDOG

9.1 INTRODUCTION

A programmable 16-bit down counter with an 8-bit prescaler is included in the ST9 Core. This Timer can be programmed to be used as a general purpose 16-bit Timer, with associated input and output pins for timing functions, or as a Watchdog Timer offering security against possible processor malfunctions due to hardware or software failures.

The Timer/Watchdog functions can use inputs from an external pin and an Alternate Function output of an I/O Port. The Input pin can be used in one of the four programmable input modes:

- event counter,
- gated external input mode,
- triggerable input mode,
- retriggerable input mode.

The output pin can be used to generate a square or a Pulse Width Modulated signal.

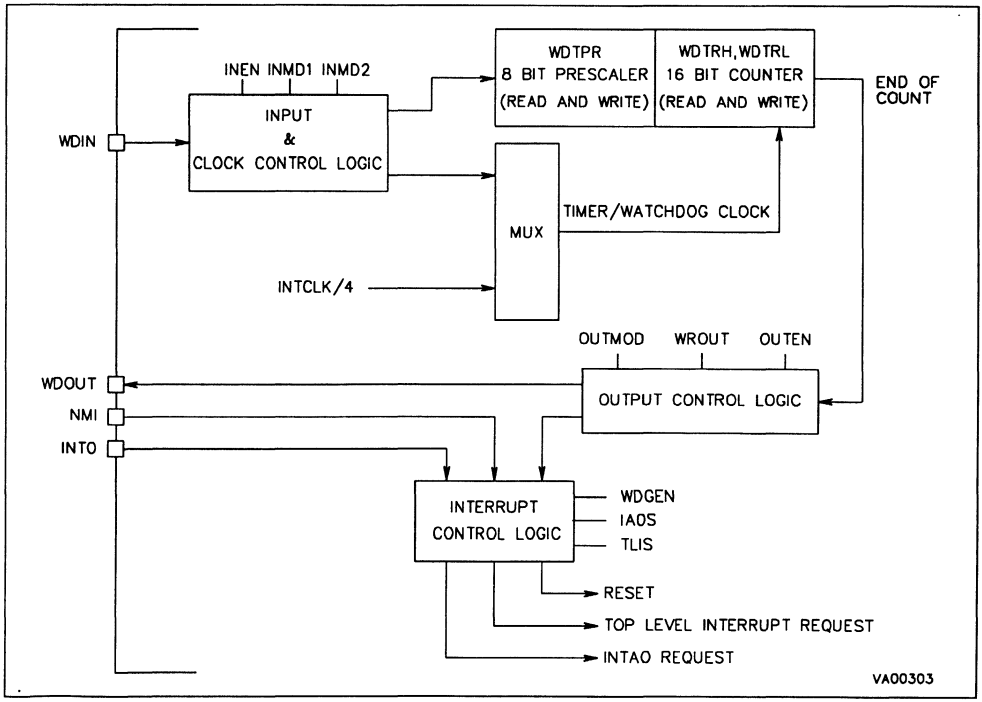
An interrupt generated by the unit (when running as a 16-bit Timer/counter and not as Watchdog) can be used as a Top Level Interrupt or as an interrupt source connected to channel A0 of the external interrupt structure (replacing the INT1 interrupt input).

The clock for the counter can be driven either by an external clock or an internal clock equal to INTCLK divided by 4.
When using an external 24MHz crystal (INTCLK = 12MHz), the End Of Count rate is:

5.59 sec. for Max. Count (Timer Const. = FFFFh, Prescaler Const. = FFh)

333 nsec. for Min. Count (Timer Const. = 0000h, Prescaler Const. = 00h)

Figure 9-1. Block Diagram



9.2 FUNCTIONAL DESCRIPTION

9.2.1 Timer/Counter Input Modes

Setting the Input Enable (INEN) bit enables the input mode which is selected via the INMD1 and INMD2 bits. When INEN is reset to zero, the input section is disabled and the values of INMD1 and INMD2 are don't-care.

Event Counter Mode
(INMD1 = "0", INMD2 = "0")

The Timer is driven by the signal applied to the input pin which acts as an external clock. The unit works therefore as an event counter. The event is a high to low transition of the input signal.
Spacing between trailing edges should be at least 350ns (i.e. the maximum Watchdog Timer input frequency is 2.9MHz with INTCLK = 12MHz).

Gated Input Mode
(INMD1 = "0", INMD2 = "1")

The Timer uses the Watchdog internal clock (INTCLK divided by 4) and starts and stops the Timer according to the input pin. When the status of the Input pin is High the Timer Watchdog count operation proceeds, and when Low, counting is stopped.

Retriggerable Input Mode
(INMD1 = "1", INMD2 = "1")

A Timer/Watchdog start is caused by:
a) a set of the Start-Stop bit, or
b) a High to Low (low trigger) transition on the input pin.

In order to stop the Timer, it is only necessary to reset the Start-Stop bit to zero.

Triggerable Input Mode
(INMD1 = "1", INMD2 = "0")

In this mode when the Timer is running (TIMER/WATCHDOG internal clock), a High to Low transition of the input pin causes the counting to start from the initial value. When the Timer is stopped (ST_SP bit equal to zero), a High to Low transition of the input pin has no effect.

9.2.2 Timer/Watchdog Output Modes

OUTPUT modes are selected using 2 bits of WDTCR (R251): OUTEN (Output Enable) and OUTMD (Output Mode).

When OUTMD = "0", the Timer outputs a signal with a frequency equal to half the End Of Count repetition rate. With INTCLK = 12MHz, this allows generation of a square wave with a period ranging from 666ns to 11.18 seconds.

The value of the WROUT bit is transferred to the output pin at the End Of Count and the value is held until the next End of Count when OUTMD = "1". This allows the user to generate PWM signals, by modifying the status of WROUT between End of Count events, based on software counters decremented on the Timer/Watchdog interrupt.

OUTEN = "1" enables the output function selected via OUTMD

When OUTEN = "0", the output is disabled and the output pin is held at a "1" level to allow several alternate functions on the same pin.

9.2.3 Timer/Counter Control

Start/Stop

ST_SP (WDTCR.7) enables down-counting. An instruction which sets this bit will cause the Timer to start at the beginning of the following instruction. Resetting this bit will stop the counter.

If the counter is stopped and restarted, counting will resume from the last value unless a new constant has been entered in the Timer registers. A new constant can be written with the counter running. The new value will be loaded at the following End Of Count (EOC).

WARNING: *In order to prevent incorrect counting of the Timer/Watchdog, the prescaler (WDTPR) and counter (WDTRL, WDTRH) registers must be initialised before the starting of the Timer/Watchdog. If this is not done, counting will start with the reset (un-initialised) values.*

Single/Continuous Mode

SINGLE MODE: At End Of Count the Timer stops, reloads the constant, and resets the Start/Stop bit (WDTCR.6) (user may check the current status by reading this bit). Restarting is done by setting the Start/Stop bit. Note that the Timer constant is reloaded only if it has been modified during the stop period.

CONTINUOUS MODE: At End Of Count the counter automatically reloads the constant and restarts. It is stopped only if the Start/Stop bit is reset. This Mode bit can be written with the Timer stopped or running. It is possible to toggle the S_C bit and start the counter with the same instruction.

FUNCTIONAL DESCRIPTION (Continued)

9.2.4 Timer/Watchdog Mode

In this mode (WDGEN = "0") the counter generates a fixed time basis. When End Of Count is reached the Timer generates a system Reset.

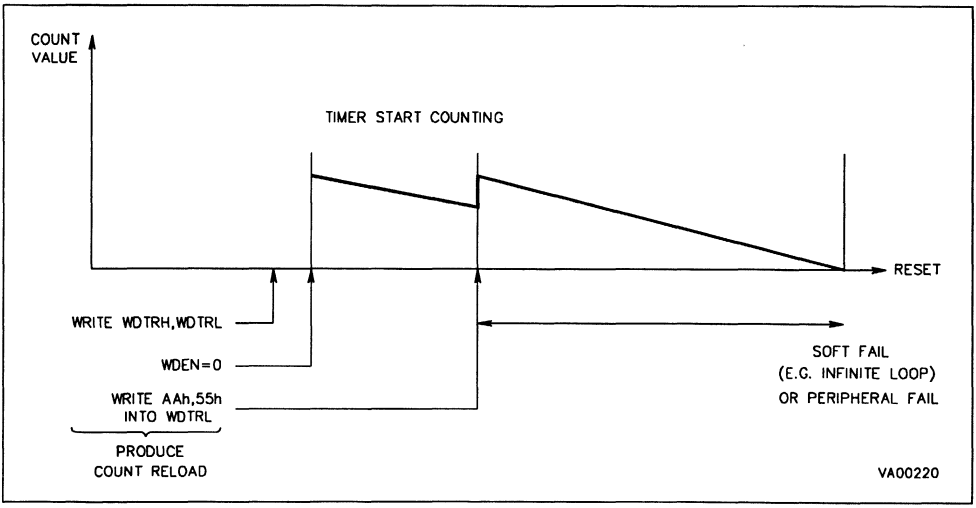
The time base is user-defined and must be written in the Timer registers before entering Watchdog mode. In Watchdog mode it is possible to modify only the Prescaler Constant. This new value will be loaded when the counter restarts.
Resetting WDGEN (bit 6 of the Wait Control Register) causes the counter to start regardless of the value of the Start-Stop. In order to prevent a system reset the sequence AAh, 55h should be entered in WDTLR (Watchdog Timer register low). Once the writing of 55h has been performed the Timer reloads the constant and counting restarts from the preset value.

The minimum time between the writing of the AAh and 55h codes is zero, i.e. the writing is sequential, and the maximum time is given by the Watchdog timeout period.

In Watchdog-mode a `halt` instruction is regarded as illegal. Execution of the `halt` instruction stops further core execution by the CPU and interrupt acknowledgment, but does not stop INTCLK or CPUCLK or the Watchdog Timer, which will cause a System Reset when reaching the End of Count. Furthermore ST_SP, S_C and input mode selection bits are "don't-care". Hence regardless of their status, the counter always runs in Continuous Mode driven by the internal clock.

The Output mode should not be enabled since that particular mode of operation is meaningless.

Figure 9-2. Timer /Watchdog in Watchdog Mode



9.3 TIMER/WATCHDOG INTERRUPT

When enabled, the Timer/Watchdog will issue an interrupt request at every End Of Count.

A pair of control bits, IA0S (EIVR.1, Interrupt A0 selection bit) and TLIS (EIVR.2, Top Level Input Selection bit) allow the selection of 2 interrupt sources (the Timer/Watchdog End of Count or an external pin) in two different ways, as a top level non maskable interrupt (Software Reset) or as a source for channel A0 of the external interrupt logic.

In the Watchdog mode the End Of Count always causes a system reset.

A block diagram of the interrupt logic is given in Figure 9-3 (Note: software traps can be generated by setting the appropriate interrupt pending bit):

The following table shows all the possible configurations of the interrupt/reset sources which involve the Timer/Watchdog:

Figure 9-3. Interrupt Sources

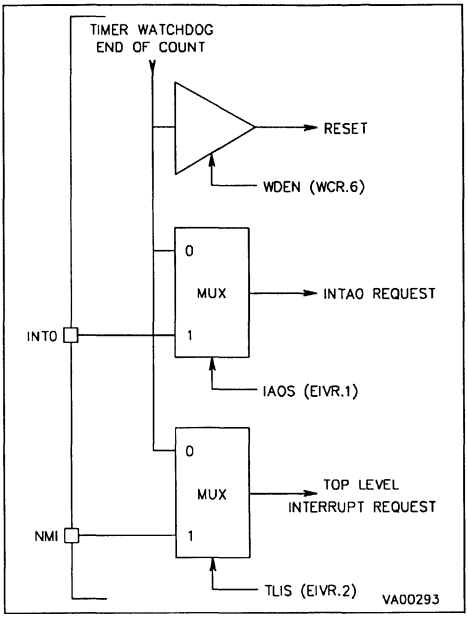


Table 9-1. Interrupt Configuration

Control Bits			Enabled Sources			Watchdog Timer Status
WDGEN	IAOS	TLIS	Reset	INTA0	Top Level	
0	0	0	WDG/Ext Reset	SW TRAP	SW TRAP	Watchdog
0	0	1	WDG/Ext Reset	SW TRAP	Ext Pin	Watchdog
0	1	0	WDG/Ext Reset	Ext Pin	SW TRAP	Watchdog
0	1	1	WDG/Ext Reset	Ext Pin	Ext Pin	Watchdog
1	0	0	Ext Reset	Timer	Timer	Timer
1	0	1	Ext Reset	Timer	Ext Pin	Timer
1	1	0	Ext Reset	Ext Pin	Timer	Timer
1	1	1	Ext Reset	Ext Pin	Ext Pin	Timer

Note.
WDG = Watchdog function
SW TRAP = Software Trap

9.4 TIMER/WATCHDOG REGISTERS

The Timer/Watchdog has 4 registers mapped into Group F, Page 0 of the Register File.

WDTHR (R248): Timer/Watchdog Counter High Register
WDTLR (R249): Timer/Watchdog Counter Low Register
WDTPR (R250): Timer/Watchdog Prescaler Register
WDTCR (R251): Timer/Watchdog Control Register

Three additional control bits are mapped in the following registers of Page 0:

- watchdog mode enable, WCR.6
- top level interrupt selection, EIVR.2
- interrupt A0 channel selection, EIVR.1

Note: The registers containing these bits also contain other functions. Only the bits relevant to the operation of the Timer/Watchdog are shown here.

Counter Registers

This 16 bit register is used to load the 16 bit counter value. The registers can be read or written "on the fly".

WDTHR R248 (F8h) Page 0 Read/Write
Timer/Watchdog Counter Register, High byte

Reset value: undefined

7							0
R15	R14	R13	R12	R11	R10	R9	R8

WDTLR R249 (F9h) Page 0 Read/Write
Timer/Watchdog Counter Register, Low byte.

Reset value: undefined

7							0
R7	R6	R5	R4	R3	R2	R1	R0

WDTPR R250 (FAh) Page 0 Read/Write
Timer/Watchdog Prescaler Register

Reset value: undefined

7							0
PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0

b7-b0 = **PR7-PR0:** *Timer/Watchdog Prescaler.* The value stored in this Register is used to select the prescaling factor from 1 (loading 00h) to 256 (loading FFh).

WARNING. *In order to prevent incorrect counting of the Timer/Watchdog, the prescaler (WDTPR) and counter (WDTRL, WDTRH) registers must be initialised before the starting of the Timer/Watchdog. If this is not done, counting will start with the reset (un-initialised) values.*

WDTCR R251 (FBh) Page 0 Read/Write
Timer/Watchdog Control Register

Reset value: 0001 0010 (12h)

7							0
ST_SP	S_C	INMD1	INMD2	INEN	OUTMD	WROUT	OUTEN

b7 = **ST_SP:** *Start/Stop Bit.* Setting this bit to a "1" starts the counting operation (see Warning above). When this bit is "0", the counter is stopped (reset status)

b6 = **S_C:** *Single/Continuous.* When this bit is set, the counter operates in Single Count Mode. Continuous Mode is set when this bit is "0"

b5-b4 = **INMD1, INMD2:** *Input mode selection bits.*

b3 = **INEN:** *Input Enable.* This bit enables ("1") and disables ("0") the input section

b2 = **OUTMD:** *Output Mode.* When this bit is "1", and the output is enabled, the value of WROUT is transferred to the output pin on every End Of Count. When "0", the output is toggled on every End of Count

b1 = **WROUT:** *WROUT bit.* The status of this bit is transferred to the Output pin when OUTMD = "1", it is user definable to allow PWM output (at reset WROUT = "1")

b0 = **OUTEN:** *Output Enable bit.* The output is enabled by setting this bit to "1", and disabled by resetting to "0"

TIMER/WATCHDOG REGISTERS (Continued)

WCR R252 (FCh) Page 0 Read/Write

Wait Control Register

Reset value: 0111 1111 (7Fh)

7							0
X	WDGEN	X	X	X	X	X	X

b6 = **WDGEN:** *Watchdog Enable Bit (active low).* Resetting this bit to zero via software enters the Watchdog mode. Once reset, it cannot be set to "1" by the user program. At system reset, the Watchdog mode is disabled

EIVR R246 (F6h) Page 0 Read/Write

External Interrupt Vector Register

Reset value: xxxx 0110 (X6h)

7							0
X	X	X	X	X	TLIS	IAOS	X

b2 = **TLIS:** *Top Level Input Selection bit.* This bit selects the Top Level interrupt source. When "0", the Top Level interrupt source is the Watchdog/Timer end of count, when = "1", it is the external pin NMI.

b1 = **IAOS:** *Interrupt channel A0 Selection Bit.* This bit allows the Timer/Watchdog interrupt to channel through the external Interrupt A0 source, allowing the setting of user-defined priority levels.

WARNING. *To avoid spurious interrupt requests, an access to the IAOS bit must be made only when the interrupt logic is disabled (i.e. after the DI instruction). It is also necessary to clear a possible interrupt pending request on channel A0 before enabling this interrupt channel. A delay instruction (e.g. a NOP instruction) must be inserted between the reset of the interrupt pending bit and the IAOS write instruction.*

10 SLICE TIMER

10.1 INTRODUCTION

The Slice Timer unit is similar in function to the Timer/Watchdog, but without the Watchdog function, providing a cost-effective solution to simple timing requirements. The Slice Timer includes a programmable 16-bit down counter and an associated 8-bit prescaler with Single and Continuous counting modes capability.

The Slice Timer uses an input from an external pin (SLIN) and an output (SLOUT) as an Alternate function of an I/O bit. SLIN can be used in one of four programmable input modes:

- event counter,
- gated external input mode,
- triggerable input mode,
- retriggerable input mode.

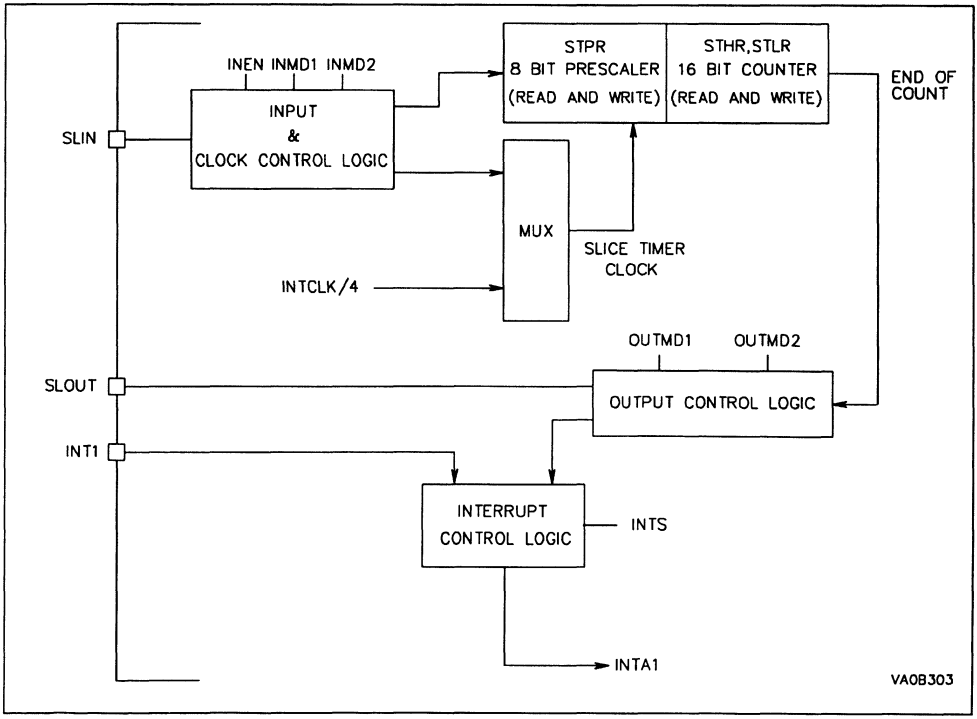
SLOUT can be used to generate a Square Wave or Pulse Width Modulated signal.

The Slice Timer is composed of a 16-bit down counter with an 8-bit prescaler. The input clock to the prescaler can be driven either by an external clock or an internal clock equal to INTCLK divided by 4.
Thus when a 24MHz crystal is used (INTCLK = 12MHz) a 3MHz maximum counting frequency can be reached or 2MHz with a 8MHz crystal and the divider by 2 disabled. The minimum and maximum counting period are:

- 5.59s for Maximum Count (Timer Const = FFFFh, Prescaler Const = FFh)

- 333ns for Minimum Count (Timer Const = 0000h, Prescaler Const = 00h)

Figure 10-1. Slice Timer Block Diagram



SLICE TIMER (Continued)

The Slice Timer End Of Count condition is able to generate an interrupt which is connected to INT1 of the external interrupt structure.

The End of Count condition is defined as the Counter Overflow, whenever 00h is reached.

10.2 SLICE TIMER FUNCTIONS

10.2.1 Timer/Counter control

Start-stop Count. The ST-SP bit (STCR.7) is used in order to start and stop counting. An instruction which sets this bit will cause the Slice Timer to start counting at the beginning of the next instruction. Resetting this bit will stop the counter.

If the counter is stopped and restarted, counting will resume from the value held at the stop condition, unless a new constant has been entered in the Slice Timer registers. during the stop period. In this case, the new constant will be loaded as soon as counting is restarted.

WARNING: In order to prevent incorrect counting of the Slice Timer, the prescaler (STPR) and counter (STLR, STHR) registers must be initialised before the starting of the timer. If this is not done, counting will start with the reset (un-initialised) values.

Single/continuous Mode. The S-C bit (STCR.6) selects between the Single or Continuous mode.

SINGLE MODE: at the End of Count, the Slice Timer stops, reloads the constant and resets the Start/Stop bit (the user programmer can inspect the timer current status by reading this bit). Setting the Start/Stop bit will restart the counter.

CONTINUOUS MODE: At the End of the Count, the counter automatically reloads the constant and restarts. It is only stopped by resetting the Start/Stop bit.

The S-C bit can be written either with the timer stopped or running. It is possible to toggle the S-C bit and start the Slice Timer with the same instruction.

10.2.2 Slice Timer Input Modes

Bits INMD2, INMD1 and INEN are used to select the input modes. Input Enable (INEN) bit enables the input mode selected by the INMD2, INMD1 bits. If the input is disabled (INEN = "0"), the values of INMD2 and INMD1 are not taken into account. In this case, this unit acts as a 16-bit timer (plus prescaler) directly driven by INTCLK/4 and transitions on the input pin have no affect.

Event Counter Mode (INMD1 = "0", INMD2 = "0")

The Slice Timer is driven by the signal applied to the input pin (SLIN) which acts as an external clock. The unit works therefore as an event counter. The event is a high to low transition on SLIN.
Spacing between trailing edges should be at least the period of INTCLK divided by 4 (i.e. the maximum Slice Timer input frequency is 3MHz with INTCLK = 12MHz).

Gated Input Mode (INMD1 = "0", INMD2 = "1")

The Timer uses the internal clock (INTCLK divided by 4) and starts and stops the Timer according to the state of SLIN pin. When the status of the SLIN is High the Slice Timer count operation proceeds, and when Low, counting is stopped.

Triggerable Input Mode
(INMD1 = "1", INMD2 = "0")

The Slice Timer is started by:

a) a set of the Start-Stop bit, AND
b) a High to Low (low trigger) transition on SLIN.

In order to stop the Slice Timer in this mode, it is only necessary to reset the Start-Stop bit.

Retriggerable Input Mode
(INMD1 = "1", INMD2 = "1")

In this mode, when the Slice Timer is running (with internal clock), a High to Low transition on SLIN causes the counting to start from the last constant loaded into the STLR/STHR and STPR registers. When the Slice Timer is stopped (ST-SP bit equal to zero), a High to Low transition on SLIN has no effect.

10.2.3 Slice Timer Output Modes

OUTPUT modes are selected using 2 bits of STCR: OUTMD1 and OUTMD2.

No Output Mode (OUTMD1 = "0", OUTMD2 = "0")

With this setting the Slice Timer output is disabled and the output pin is held at a "1" level to allow several alternate functions on the same pin.

Square Wave Output Mode (OUTMD1 = "0", OUTMD2 = "1")

The Slice Timer toggles the state of the SLOUT pin on every End Of Count condition. With INTCLK = 12MHz, this allows generation of a square wave with a period ranging from 666ns to 11.18 seconds.

SLICE TIMER (Continued)

PWM Output Mode (OUTMD1 = "1")

The value of the OUTMD2 bit is transferred to the SLOUT output pin at the End Of Count. This allows the user to generate PWM signals, by modifying the status of OUTMD2 between End of Count events, based on software counters decremented on the Slice Timer interrupt.

10.2.4 Interrupt Selection

The Slice Timer may generate an interrupt request at every End of Count.

The STCR bit 2 (INTS) selects the interrupt source between the Slice Timer interrupt and the external interrupt INT1. Thus the Slice Timer Interrupt uses the INT1 interrupt channel and takes the priority and vector of the INTA1 external interrupt channel.

If INTS is set to "1", the Slice Timer interrupt is disabled; otherwise, an interrupt request is generated at every End of Count.

Care must be taken when disabling the Slice Timer Interrupt as a rising edge may be generated on the INTA1 channel, causing an unwanted interrupt.

10.2.5 Slice Timer Registers

This unit has 4 registers mapped into the page 0Bh in Group F of the Register File:

Register Address	Register	Function
F0	STH	Counter High-Byte Register
F1	STL	Counter Low-Byte Register
F2	STP	Slice Timer Prescaler Register
F3	STC	Slice Timer Control Register
F4-FF	—	Reserved

10.2.6 Register Description

STH R240 (F0h) Page 0B Read/Write
Counter High-Byte Register

Reset value: undefined

7							0
ST.15	ST.14	ST.13	ST.12	ST.11	ST.10	ST.9	ST.8

b7-b0 = **ST.15-ST.8:** *Counter High-Byte.*

STL R241 (F1h) Page 0B Read/Write
Counter Low-Byte Register

Reset value: undefined

7							0
ST.7	ST.6	ST.5	ST.4	ST.3	ST.2	ST.1	ST.0

b7-b0 = **ST.7-ST.0:** *Counter Low-Byte.* Writing to the STH and STL registers allows the user to enter the Slice Timer constant, while reading provides the counter current value. Thus it is possible to read the counter on-the-fly.

STP R242 (F2h) Page 0B Read/Write
Slice Timer Prescaler Register

Reset value: undefined

7							0
STP.7	STP.6	STP.5	STP.4	STP.3	STP.2	STP.1	STP.0

b7-b0 = **STP.7-STP.0:** *Prescaler.* The Prescaler value for the Slice Timer is programmed into this register. When reading the STPR register, the returned value corresponds to the programmed data instead of the current data.

STC R243 (F3h) Page 0B Read/Write
Slice Timer Control Register

Reset value: 000x x1xx

7							0
ST-SP	S-C	INMD1	INMD2	INEN	INTS	OUTMD1	OUTMD2

b7 = **ST-SP:** *Start-Stop Bit.* Setting ST-SP to "1" starts the counting operation. Writing "0" stops the Slice Timer.

b6 = **S-C:** *Single-Continuous Mode Select.* Setting S-C to "1" sets the Slice Timer to Single Mode. Writing "0" sets the Continuous Mode (Reset Status)

b5-b4 = **INMD1, INMD2:** *Input Mode Selection.* These bits select the Input functions as shown in the preceding text, when enabled by INEN.

b3 = **INEN:** *Input Enable.* INEN must be set to "0" (Input section disabled).

b2 = **INTS:** *Interrupt Selection.* Setting INTS to "1" disables the Slice Timer interrupt (the Reset status).
Writing "0" enables the Slice Timer interrupt on channel INT1.

b1-b0 = **OUTMD1, OUTMD2:** *Output Mode Selection.* These bits select the output functions as described in the preceding text.

NOTES

11 ON SCREEN DISPLAY

11.1 INTRODUCTION

The ST9 On-Screen Display (OSD) is used to display lines of characters under control of the ST9 CPU on a Video Display screen (for example close caption displays on a television). The screen insertion of the displayed characters is fully synchronised by the vertical and horizontal TV synchronization signals. The OSD generates the Red, Green, Blue and Fast blanking video signals.

The OSD displayable characters are located into the character ROM memory. The set of 128 different characters is fully user definable by ROM mask option.

To give more flexibility for using the OSD, a buffered one-row RAM architecture is used.

Only one row of characters is described at a time within the one-row RAM memory. This one-row Display RAM contains the description of the currently displayed row: row and word attributes, horizontal placement and character codes which will be converted into pixel information issued from the character ROM memory.

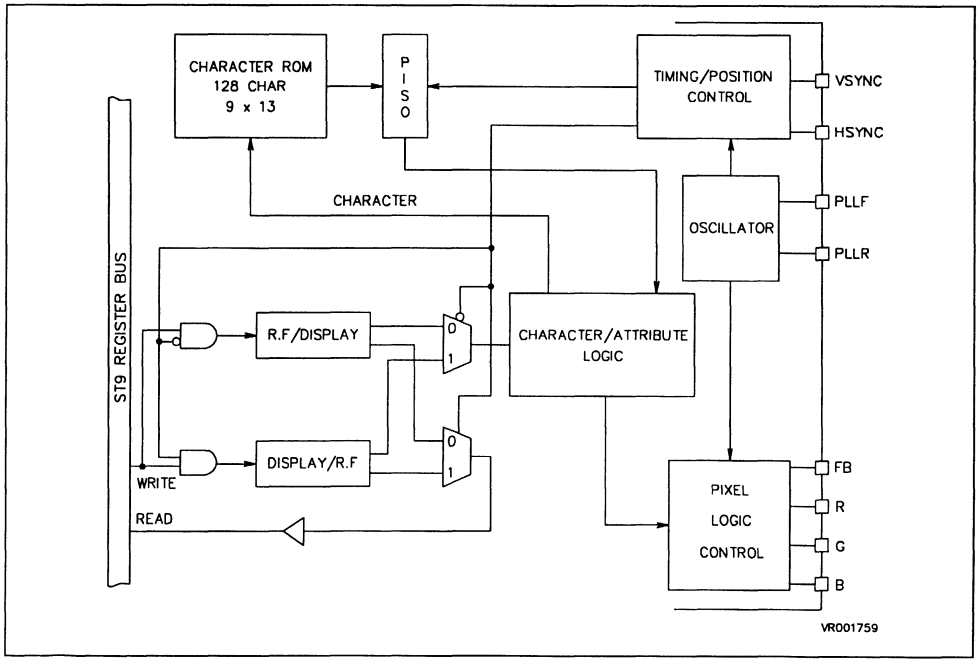
During the time the Display RAM is displaying on the screen, the CPU can prepare the content of the next row of characters by re-writing the OSD RAM registers. At the time the next row must be displayed the RAM register content is swapped to the one-row Display RAM, allowing the CPU to prepare the further row content and location.

An interrupt request may be generated each time the swap occurs (i.e. each time the one-row RAM content swapped to the display function).

The interrupt request uses the INT5 channel, taking the priority and mask of INTC1 and the vector programmed in EIVR.

Other information such as the vertical placement of the next row of characters is given as a vertical ad-

Figure 11-1. OSD Block Diagram



ON SCREEN DISPLAY (Continued)

dress by programming the Event Line register. The real-time vertical position is calculated by the Scan Line counter which counts the TV horizontal synchronisation pulses. Then, at the time the Scan Line counter matches the Event Line value, the swap mechanism is activated.

This architecture allows the CPU to perform vertical scrolling of the rows by software.

11.1.1 Format

Each row is able to display 34 characters in small size mode.

The row RAM size is 36 bytes. The two first locations are not displayed but are used as attribute locations.

Each character is composed by a 9x13 pixel matrix.

Each pixel is at least 2 clock periods wide and 2 lines high. The pixel clock is generated by an internal controlled oscillator, whose frequency is locked to the horizontal synchronisation signal as follows:

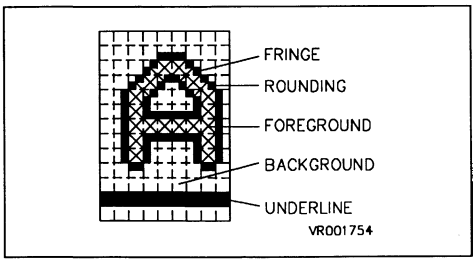
Fosc = 896 x F(HSYNC)

The pixel pattern of the character is stored in the character ROM.

The colour controller displays the foreground colour when a "1" is read from the ROM or displays the background colour when a "0" is read.

These attributes can be modified by the rounding and fringe logic.

Figure 11-2. Character Formats



As shown in Figure 11-2 the rounding mechanism is done by adding a half-dot to smooth the diagonal lines. The half-dot is painted as foreground.

The fringe is a half-dot wide black border. Fringe and rounding may be turned on by setting word attribute bits.

All the screen area outside of the displayed rows of characters is displayed using the border colour.

The colour of background, foreground and border are sent to the OSD controller as attributes.

11.1.2 Attributes

The attributes of the OSD are serial: Two kinds of attributes are available:

- Row attributes which apply to the entire row.

- Serial attributes which apply until changed, or to the end of the line.

Row attributes are defined by two registers. These 2 registers are copied on each swap (when the RAM registers are switched to the one-row Display RAM). The modifications made in the row attribute registers are active during the next row and the following if no new write occurs.

The row attribute registers are initialized by the CPU reset condition.

Row attributes 1 register:

HDCS R244 (F4h) Page 2A Read/Write
Horizontal Delay/Character Size Register

Reset value: 0000 0000b (00h)

7							0
SH	SW	HD5	HD4	HD3	HD2	HD1	HD0

The HDCS Register sets the character size for the display row, and the horizontal delay for positioning the start of the row on the screen relative to the start of the raster line.

b7-b6 = **SH, SW:** *Pixel Size.* These two bits are encoded to select the pixel size for the displayed characters.

Pixel Size for Displayed Characters (Single Scan)

SH	SW	Pixel Height Horizontal lines/field	Pixel Width		Max. Char. /Row
			#Clock	H	
0	0	1	2	1H	34
0	1	2	2	2H	34
1	0	2	3	1.5H	24
1	1	2	4	2H	18

b5-b0 = **HD5-HD0:** *Horizontal delay.* The starting position on the video line can be offset by programming the HD value. This value is loaded into a down-counter using the OSD clock as input at the

ON SCREEN DISPLAY (Continued)

beginning of the line. When the counter reaches 0, the display will start (beginning with the two undisplayed characters).

The total delay from the active edge of HSYNC to the position of the first displayed character is equal to:

86 x T_{osc} + [HDel x (HDRES+1)+18] x T_{pixel}

Where T_{pixel} (width of one pixel) is:

$2T_{osc}$ for SH, SW= 0, X

$3T_{osc}$ for SH, SW= 1, 0

$4T_{osc}$ for SH, SW= 1, 1

HD_{RES} = "0" for small resolution (refer to the Enable Register description)

Row attributes 2 register:

AR R245 (F5h) Page 2A Read/Write
Active Range Register

Reset value: 0000 0000b (00h)

7							0
RS3	RS2	RS1	RS0	RE3	RE2	RE1	RE0

b7-b4 = **RS3-RS0:** *Active Range Start Value.*

b3-b0 = **RE3-RE0:** *Active Range Stop Value.*

These two values are compared to the vertical pixel character counter value. The RS range is 0 to 12 and RE range is 1 to 13.

The counter is reset to 0 at the beginning of each row and counts each vertical pixel row up to 12.

If the value of the counter is outside the range (less than RS3-RS0 or greater than RE3-RE0) the OSD displays the border attributes, otherwise the display is normal.

For example:

If RS = 0 and RE = 1, only the first line of the row is displayed.

This feature is useful for the software generated vertical smooth scrolling, a row of characters can appear or disappear line by line.

11.1.3 Serial Attributes:

The serial attributes registers describe how the current row is built. There are stored within the displayed line buffer in page 28h: F0h-FFh, page 29h: F0h-FFh and page 2Ah: F0h-F3h (refer to the OSD Memory Map).

The serial attribute is sent to the OSD inside the display RAM. When a valid serial attribute code is encountered the corresponding serial attribute register is loaded with the new value, and remains in effect until the end of the row, or the next serial attribute.

If the most significant bit of the RAM location is "0", the seven last significant bits have the meaning of the code of the character, if it is "1", the byte contains a serial attribute.

In this case the OSD displays a space and stores the new attributes. These attributes will remain active until a new value is encountered.

The first two locations of the row RAM have a special behaviour: They are displayed as border, but if they contain a valid serial attribute (MSB = "1") the corresponding attribute register is modified.

Two sets of attributes can be stored using bit 6 as address bit:

If b6 = "0" the "foreground attributes" are modified.

If b6 = "1" the "additional attributes" are modified.

Character code byte:
(b7 = "0")

7							0
0	C6	C5	C4	C3	C2	C1	C0

Character 00h displays the border colour. Normally the border colour begins at the first column of the character matrix and is a full 9 pixels width. However, there is an exception to this rule. If the character immediately preceding the 00h is in the range of 01h-7Fh, and the italics word attribute is on, the previous character will extend half way into the 00h. In this case, the border colour will begin in the middle of the character matrix and extend to the right edge of the matrix. The right edge of character 00h is never extended into the next character.

Characters 01h-7Fh are the printable characters. Normally they will display the pixel pattern defined in the character ROM, and will be 9 pixels wide. However, there is an exception to this rule. If the italics serial attribute is on, the character will be slanted to the right.

Each character in a displayed word will be 9 pixels wide except for the last. This will have the matrix extended to the right, half way into the next character. This is true regardless of what the next character is, unless it is the last character to be displayed. In this case, it will be truncated to 9 pixels wide.

The serial attributes: flash, underline, round, and fringe also effect these characters.

ON SCREEN DISPLAY (Continued)

Foreground attributes byte:
(b7 = "1", b6 = "0")

7							0
1	0	FLA	ITA	UND	FOR	FOB	FOG

b6 = **FLA:** *Flash Enable.* FLA = "1" forces the flashing mode for the following word. According to the value of the "flash on" bit (located into the Enable register), the word is displayed as in normal mode (FON = "0") or displayed as a space with background colour (FON = "1").

Underline also flashes.

b5 = **ITA** *Italic Enable.* ITA = "1" forces the following characters to be displayed with the Italic attribute.

Italic is displayed by applying a slant to the character. The slope is equal to 2.5 pixels horizontally for the 13 vertical pixels. The horizontal shift for each line is the following:

lines 0-1 : 2.5p lines 6-7 : 1p
lines 2-3 : 2p lines 8-9 : 0.5p
lines 4-5 : 1.5p lines 10-12 : 0p

The mandatory spaces needed to set and reset the italic attribute are used to solve the border problems between italic and the normal character shape.

b3 = **UND:** *Underline enable.* UND = "1" forces the underlining of the following word. The 11th (relative to 0) row of character pixels is painted as foreground. The Underline affects only words (spaces are not underlined).

b2-b0 = **FOR, FOG, FOB:** *Foreground colour select.* FOR, FOG and FOB bit control the RGB output for the foreground colour.

FOR	FOG	FOB	Foreground colour
0	0	0	Black
0	0	1	Blue
0	1	0	Green
1	0	0	Red
0	1	1	Cyan
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

Additional attributes byte:
(b7 = "1", b6 = "1")

7							0
1	1	ROU	FR	BGS	BGR	BGB	BGG

b5 = **ROU:** *Rounding enable.* ROU = "1" enables Rounding of the following characters, ROU = "0" disables character Rounding.

b4 = **FR:** *Fringe enable.* FR = "1" enables the Fringe effect for the following characters, FR = "0" disables the Fringe.

b3 = **BGS:** *Transparency select.* BGS = "1" sets the background to a solid colour defined by BGR, BGB, BGG. If BGS = "0", the background is transparent.

The transparency of the background is modified in the middle of the space containing the serial attribute.

b2-b1 = **BGR, BGB, BGG:** *Background colour select.* BGR, BGB and BGG bit control the RGB output for the background colour. The background colour modification occurs in the middle of the serial attribute space.

The colours generated follow the same coding as shown for FOR, FOG and FOB.

Characters 80h-FFh are control characters, but will be displayed as background colour only. They always display the first half of the character matrix with the previous background colour, and then the second half with the current background colour. If they are immediately preceded by an italic printing character, the italic character will extend half-way into the control character. The right edge of control characters is never extended into the next character. The serial attributes: flash, underline, rounding, and fringe have no effect on control characters.

ON SCREEN DISPLAY(Continued)

Figure 11-3. Italic Attributes

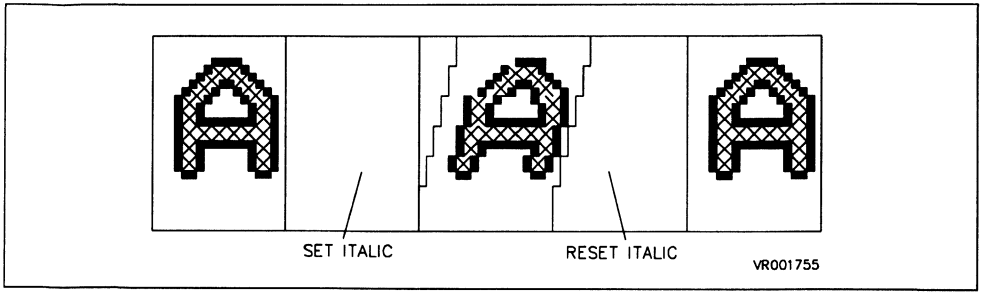
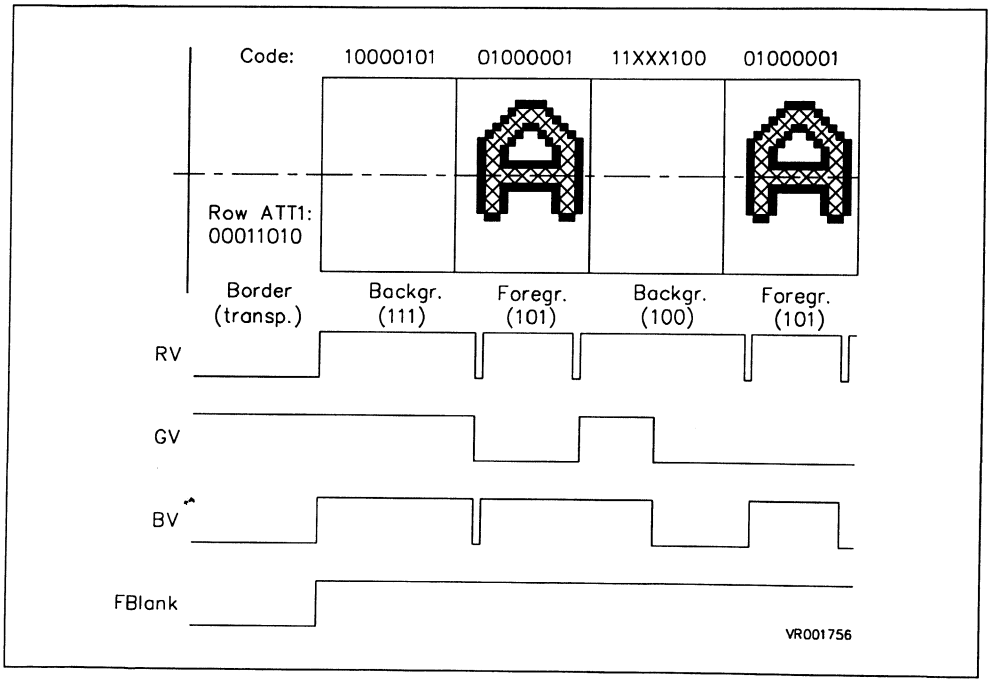


Figure 11-4. Colour Outputs



ON SCREEN DISPLAY(Continued)

11.1.4 Status Registers

The status registers give the programmer a real time status of OSD functions. This data is useful to synchronize the software with the display.

The status registers are READ-ONLY registers.

SL R249 (F9h) Page 2A Read-only
Scan Line Register

Reset value: 0000 0000b (00h)

7							0
SL7	SL6	SL5	SL4	SL3	SL2	SL1	SL0

b7-b0 = SL7-SL0: *Scan Line Counter.* The nine bit counter SL8-SL0 indicates the vertical position of the beam. This counter starts at 0 at the top of the screen and is incremented after HSY in case of normal scan mode, and after 2 HSY in case of double scan mode (2H).

FB R250 (FAh) Page 2A Read-only
Flag Bit Register

Reset value: 0000 0000b (00h)

7							0
SL8	VSY	HSY	0	VP3	VP2	VP1	VP0

b7 = **SL8:** *Scan Line bit 8.* SL8 is the MSB (ninth) bit of the Scan Line counter (see above).

b6 = **VSY:** *Vertical Sync active.* This bit is a flag activated (set to "1") during the vertical fly-back, information being issued from the VSYNC pad controlled by the programming of the Delay register and the VSYNC polarity (refer to the Border Color register).

b5 = **HSY:** *Horizontal Sync active.* This bit is a flag activated (set to "1") on the starting edge of the HSYNC pad signal. HSY remains active during 4µs in single scan (2µs in double scan mode).

b3-b0 = **VP3-VP0:** *Vertical Sync Phase.* This nibble gives the phase value between the starting edge of the internal VSYNC signal and the starting edge of the first internal HSYNC signal that follows VSYNC. Phase(VSYNC-HYNC) = VP x 4µs

Note: VP3 can be used as a field polarity flag (refer to the Delay register).

11.1.5 Control Registers

The control registers are used by the programmer to control the behavior of the OSD.

BCOL R251 (FBh) Page 2A Read/Write
Border Color Register

Reset value: 0000 0000b (00h)

7							0
HPOL	VPOL F	BPOL	RGBPOL	BOS	BOR	BOB	BOG

b7 = **HPOL:** *HSYNC pad polarity.* HPOL = "0" for positive polarity (i.e. HSYNC = "1" during the line synch pulse).

b6 = **VPOL:** *VSYNC pad polarity.* VPOL = "0" for positive polarity (i.e. VSYNC = "1" during the frame synch pulse).

b5 = **FBPOL:** *Fast blanking polarity.* FBPOL = "0" for positive polarity.

b4 = **RGBPOL:** *RGB outputs polarity.* RGBPOL = "0" for positive polarities.

b3 = **BOS:** *Border Transparency Select.* When BOS = "1" the border is solid (displayed with the colour set by BOR, BOB, BOG).

When BOS = "0" the border is transparent.

b2-b0 = **BOR, BOB, BOG:** *Border colour.* Bor, Bob, Bog bits program the value of the respectively R/B/G outputs when displaying the border.

ON SCREEN DISPLAY(Continued)

EN R247 (F7h) Page 2A Read/Write
Enable Register

Reset value: 0000 000xb (0xh)

7							0
0	TE	FON	OSDE	PLLE	1H/2H	HDRES	EL8

b7 = Read only to "0"

b6 = **TE:** *Transfer Enable.* It is used to enable (TE = "1") or disable (TE = "0") the swap of the RAM registers content to the display RAM each time the Scan Line counter content matches the Event Line register value.

An interrupt request pulse is generated and forwarded to the core each time the match occurs independently to the value of TE.

If TE = "0" no transfer occurs (no swap) and the screen will display border colour.

b5 = **FON:** *Flash On.* FON= "1" Flash on: The characters with the flash attribute are displayed as space. FON = "0" The flash attribute has no effect on the characters. This bit is used to control the flashing period by software.

b4 = **OSDE:** *OSD Enable.* When set it will start the OSD function and allow displaying. When reset, it will stop the OSD clock and force the Blanking ouput in its inactive state; but OSD basic functions such as: Event Line counter, interrupt request generation, Flag Bit and Delay registers are remaining active.

b3 = **PLLE:** *PLL Oscillator Enable.* If set to "1", the internal pixel oscillator is started and runs, if set to "0", the internal oscillator is stopped.

b2 = **1H/2H:** *Scan Select.* "0" for single scan, "1" for double scan (31kHz).

b1 = **HDRES:** *Horizontal Delay Resolution.* If HDRES = "0" "1 pixel" resolution for the Horizontal delay if HDRES = "1" "2 pixel" resolution.

b0 = **EL8:** *Event Line Register bit 8.* EL8 is the MSB (ninth) bit of the Event Line register (see Register EL).

Remark: The control of the video outputs (Fast blanking and RGB) is made using the standard Alternate Function I/O port features (see I/O ports chapter)

EL R246 (F6h) Page 2A Read/Write
Event Line Register

Reset value: 0000 0000b (00h)

7							0
EL7	EL6	EL5	EL4	EL3	EL2	EL1	EL0

b7-b0 = **EL7-EL0:** *Event Line Register.* The Event Line register is written by the CPU to indicate the next vertical location for displaying the row of characters. The accuracy of the position is one line in single scan mode. When the content of the scan line register matches with the content of the event line register an interrupt request pulse is generated and forwarded to the core on the INT4(INTC0) channel.

VD R248 (F8h) Page 2A Read/Write
Vertical Delay Register

Reset value: 1111 0000b (F0h)

7							0
1	1	1	1	VD3	VD2	VD1	VD0

b7-b4 = Read only to "1"

b3-b0 = **VD3-VD0** *Vertical Delay.* This programmable delay is applied on the Vertical Synchronisation input to balance the delay introduced by the scanning processor.

4 bits value programmable delay:

d= VD x 4µs (up to 60µs).

Using a quarter of line delay, the distinction between the two fields is possible:

If the active edge of the delayed vertical signal occurs in the first half of the line the parity flag is reset (even field); if it happens during the second half, the flag is set (odd field).

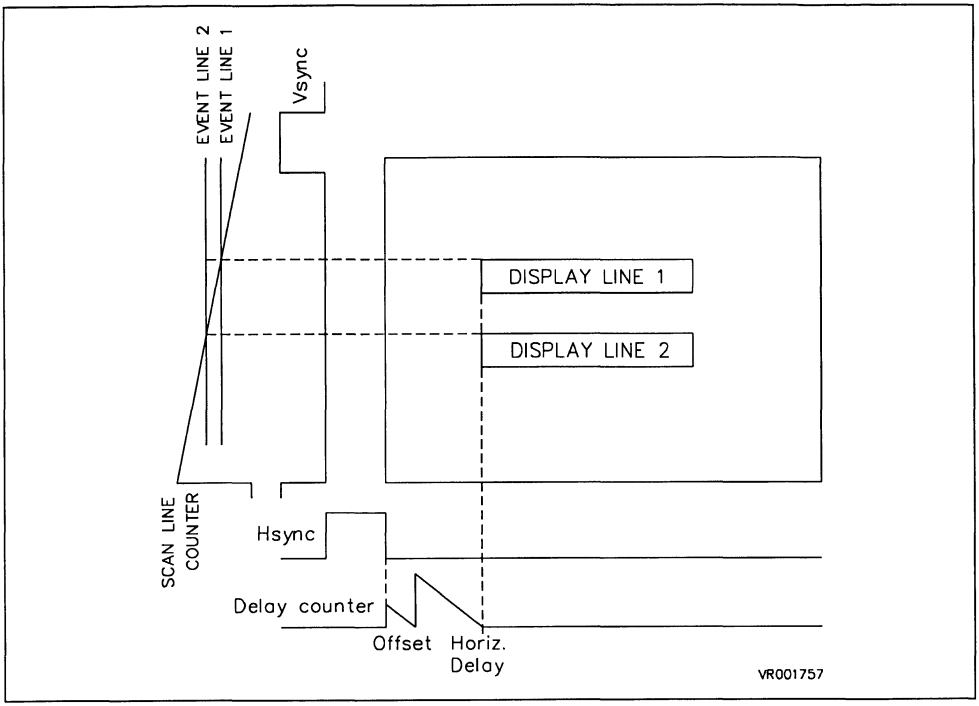
Adding an half line value to the delay of VSYNC leads to an inversion of the odd and even field of the display.

Warning: *Programming the Vertical Delay Register to FFh freezes the scan line counter, disabling any further RGB output. It is mandatory to always initialise the Vertical Delay Register and to avoid programming it to FFh.*

The value stored in this register is not affected by the row refresh mechanism. There is no need to confirm the value for each new transfer.
This value is used to compensate the delay between HSYNC and VSYNC and must not be updated each frame.

ON SCREEN DISPLAY(Continued)

Figure 11-5. OSD Line Positioning



OSD Input Timing Requirements

Symbol	Parameter	Value		Unit
		Min.	Max.	
f_{OSC}	Oscillator frequency			MHz
T_{VWL}	Vsync pulse width	4		µs
T_{HWL}	Hsync pulse width	4		µs

ON SCREEN DISPLAY(Continued)

11.1.6 RGB Blanking:

When the fast blanking FB is inactive, the RGB outputs are forced to a black level.

If FB and RGB have a positive polarity:
FB = "0" => RGB = "0"

11.1.7 OSD Memory Map

The OSD uses three pages of 16 paged registers.

Register Address.	Page 28
F0	Char. Column 1 displayed as border
F1	Char. Column 2 displayed as border
F2	Char. Column 3 First displayed character
.	.
.	.
FF	Char. Column 16
	Page 29
F0	Char. Column 17
F1	Char. Column 18
.	.
.	.
FF	Char. Column 32
	Page 2A
F0	Char. Column 33
F1	Char. Column 34
F2	Char. Column 35 33th displayed character
F3	Char. Column 36 34th displayed character
F4	Horizontal delay
F5	Active range
F6	Event line
F7	Enable register
F8	Delay register
F9	Scan line
FA	Flag bits
FB	Border colour
FC-FF	Reserved

11.1.8 OSD Interface

OSD interfaces internally with the CPU and externally with the video logic. Furthermore the OSD needs a proper clock fitting with the high frequency video signals. This is internally generated by a fully internal oscillator controlled by a PLL and a few external components as described below.

The video interface is composed by 4 output and 2 input lines. The polarities of these signals are programmed by software.

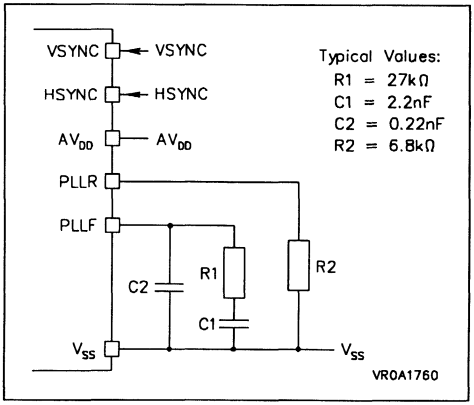
- R, G and B outputs: Color signal outputs. Positive polarity means R=G=B = "1" is white.
- FB output: Fast blanking. Positive polarity means FB = "1" if the video signal is given by the OSD, "0" if transparent (also means FB inactive) When FB is inactive the RGB signals are forced to a black level.

The I/O port bits associated to R,G,B and FB must be set to Alternate Function Output mode for these signals to appear externally.

- VSYNC input: Vertical synch pulse. Positive polarity means VSYNC = "1" during the flyback pulse.
- HSYNC input: Horizontal synch pulse. Positive polarity means HSYNC = "0" during the line (HSYNC = "1" during Horizontal retrace).

Clock pins. Three pins are dedicated to the OSD clock: PLLF, PLLR and AVDD. The clock is synchronized with the HSYNC signal with a PLL; an external loop filter must be connected to the phase comparator output PLLF. A resistor must be connected to PLLR.

Figure 11-6. OSD Oscillator



NOTES

12 IFP A/D CONVERTER

12.1 INTRODUCTION

The IFP Analog to Digital converter peripheral can convert up to 4 analog channels using the successive approximation conversion technique to a resolution of 8 bits ±2LSB. It includes four analog fully differential comparators blocks, AC coupled, plus Sample and Hold logic and a reference generator.

The internal reference (DAC) is based on the use of a binary-ratioed capacitor array. This technique provides good monotonicity by using the same ratioed capacitors as the sample capacitor.

The Single Conversion time is 5.75µs (INTCLK = 12MHz) including a Sample time of 1.5µs (typical). The 8-bit conversion uses 69 INTCLK periods for the conversion.

A continuous conversion of the selected channel is performed in Continuous mode, while in the Single mode the selected channel is only once converted and the logic then waits for a new hardware or software restart.

The start conversion event can be produced by software, by writing the START/STOP bit of the Control Logic Register (CLR), or by external hardware, by applying an external trigger on the EXTRG (negative edge sensitive).

The Data register holds the converted value data (in single or continuous mode).

A Power Down programmable bit sets the A/D converter analog section to a minimum consumption idle status.

12.1.1 Operation Modes

The two main operation modes, SINGLE and CONTINUOUS, can be selected writing "0" (reset value) or "1" into the CONT bit of the Control Logic Register:

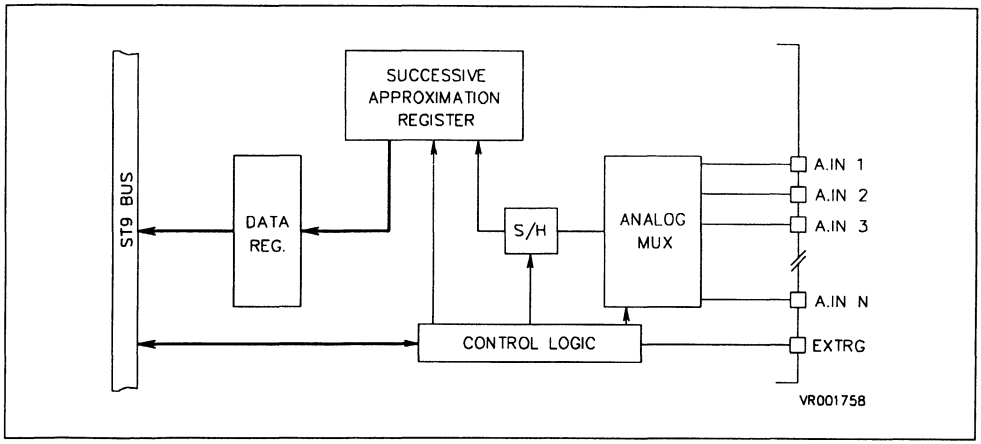
In **Single Mode** (CONT = "0" into CLR) the STR bit is forced to "0" after the end of channel `i` conversion (`i` = 1, 2 or 3); the A/D then waits for a new start event.

This mode is useful when a set of signals must be sampled at a fixed frequency imposed by an ST9 timer unit or an external generator (through the alternate synchronization feature).

A simple software routine monitoring the STR bit can be used to save the current value before a new conversion end (so to create signal samples table within internal/external memory or Register File).

In **Continuous Mode** (CONT = "1" into CLR) a continuous conversion flow is entered by a start event. After the end of conversion of channel `i` the conversion takes place again on channel `i`, repeating until the STR bit is reset by software.

Figure 12-1. IFP A/D Converter Block Diagram



IFP A/D CONVERTER (Continued)

At the end of conversion of channel i, the Data Register (DTR) is reloaded with the new conversion result and the previous value is lost.

The **Alternate Synchronization Feature** is available in both single and continuous operation modes. The external EXTRG signal can be used to synchronize the conversion start with a trigger pulse. The external events can be masked by programming the TRG bit of the Control Logic Register.

The effect of the alternate synchronization is to hardware set the STR bit, which is hardware reset at the end of each conversion only in SINGLE mode. In CONTINUOUS mode all trigger pulses, following the first one, are ignored.

The synchronization source must have a clock cycle minimum length of 83ns (at INTCLK = 12Mhz) minimum length, and a period greater (in Single mode) than the time occurring for a conversion.

If a trigger occurs when the STR bit is still "1" (conversion still in progress), it is ignored.

Note: Before enabling any A/D operation mode, it is necessary to set the POW bit of the Control Logic Register to "1" to bias the analog section of the converter at least 60µs before the first conversion; otherwise the converter functionality is locked. Setting POW to "0" is useful when the IFP A/D is not used to reduce the total power consumption. This is the reset configuration and is also automatically entered when the ST9 is in the HALT state (HALT instruction executed).

12.1.2 Register Mapping

The IFP A/D peripheral has 2 dedicated registers, mapped in the ST9 Register File page 62 (3Eh) as follows:

Register Address	Page 62 (3EH)
F0	DTR
F1	CLR
F2-FF	Reserved

12.1.3 Register Description

DTR R240 (F0h) Page 3E Read/Write
Channel i Data Register

Reset value: undefined

7							0
R.7	R.6	R.5	R.4	R.3	R.2	R.1	R.0

The results of the conversion of the selected channel is stored into the 8 bit DTR, which is reloaded with a new value every time a conversion ends.

b7-b0 = **R7-R0:** *Channel i* conversion result.

Note: Only the most significant 6 bits are characterised when the ST9 is operating during conversion (accuracy is typically 8-bit ± 2LSB). When the ST9 is in WFI mode the accuracy is increased.

IFP A/D CONVERTER (Continued)

CLR R241 (F1h) Page 3E Read/Write

Control Logic Register

Reset value: 0000 0000 (00h)

7							0
C2	C1	C0	FS	TRG	POW	CONT	STR

This 8 bit register manages the A/D logic operations. Any write operation to it will cause the current conversion to be aborted and the logic to be re-initialized to the starting configuration.

b7-b5 = **C2-C0:** *Channel Address.*

A "1" in bit Ci selects channel i conversion as follows:

C2	C1	C0	Channel Enabled
0	x	x	No channel enabled
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

b4 = **FS:** *Fast/Slow.* A logical "1" level enables a division by two of the conversion frequency, doubling the sampling time.

FS = "0" (fast mode) Single conversion time: 5.75µs at INTCLK = 12MHz

FS = "1" (slow mode) Single conversion time: 11.5µs at INTCLK = 12MHz

b3 = **TRG:** *External Trigger Enable.* A logical level "0" on this bit disables the External Trigger applied on the EXTRG pin. When TRG is set to "1", a negative (falling) edge on TRG writes a "1" into the STR bit, enabling start of conversion.

b2 = **POW:** *Power Enable.* A logical "1" level enables the A/D logic and analog circuitry. A logical level "0" disables all power consuming logic within the IFP A/D.

b1 = **CONT:** *Continuous/Single Mode Select.* A logical level "1" sets the converter in 'Continuous Mode', thus allowing a continuous flow of A/D conversions on the selected channel, starting when the STR bit is set.

A logical level "0" sets the 'Single mode'; after the current conversion end, the STR bit is hardware reset and the converter logic is put in a wait status. To start another conversion, the STR bit has to be set by software or hardware.

b0 = **STR:** *Start/Stop.* A "1" enables the start of conversion of channel i; a logical level "0" stops the converter until a new "1" is written (by hardware, when the A/D is synchronized with an external trigger, or by software).

NOTES

REGISTER MAP

2 CORE ARCHITECTURE

CICR	**R230**	(E6h)	System	Read/Write	Central Interrupt Control Register	11
FLAGR	**R231**	(E7h)	System	Read/Write	Flag Register	12
RP0	**R232**	(E8h)	System	Read/Write	Register Pointer 0	13
RP1	**R233**	(E9h)	System	Read/Write	Register Pointer 1	13
PPR	**R234**	(EAh)	System	Read/Write	Page Pointer Register	15
MODER	**R235**	(EBh)	System	Read/Write	Mode Register	15
USP	**R236**	(ECh)	System	Read/Write	User Stack Pointer High Byte	17
USP	**R237**	(EDh)	System	Read/Write	User Stack Pointer Low Byte	17
SSP	**R238**	(EEh)	System	Read/Write	System Stack Pointer High Byte	17
SSP	**R239**	(EFh)	System	Read/Write	System Stack Pointer Low Byte	17

4 INTERRUPTS

CICR	**R230**	(E6h)	System	Read/Write	Central Interrupt Control Register	36
EITR	**R242**	(F2h)	Page 0	Read/Write	External Interrupt Trigger Event Register	36
EIPR	**R243**	(F3h)	Page 0	Read/Write	External Interrupt Pending Register	36
EIMR	**R244**	(F4h)	Page 0	Read/Write	External Interrupt Mask-bit Register	37
EIPLR	**R245**	(F5h)	Page 0	Read/Write	External Interrupt Priority Level Register	37
EIVR	**R246**	(F6h)	Page 0	Read/Write	External Interrupt Vector Register	37
NICR	**R247**	(F7h)	Page 0	Read/Write	Nested Interrupt Control Register	37

5 CLOCK

MODER	**R235**	(EBh)	System	Read/Write	Mode Register	40
WCR	**R252**	(FCh)	Page 0	Read/Write	WAIT Control Register	40

8 SERIAL PERIPHERAL INTERFACE

SPIDR	**R253**	(FDh)	Page 0	Read/Write	SPI Data Register	56
SPICR	**R254**	(FEh)	Page 0	Read/Write	SPI Control Register	56
SWAP	**R255**	(FFh)	Page 0	Read/Write	SPI SWAP control Register	56

9 TIMER/WATCHDOG

WDTHR	**R248**	(F8h)	Page 0	Read/Write	Timer/Watchdog Counter Register, High byte	67
WDTLR	**R249**	(F9h)	Page 0	Read/Write	Timer/Watchdog Counter Register, Low byte.	67
WDTPR	**R250**	(FAh)	Page 0	Read/Write	Timer/Watchdog Prescaler Register	67

WDTCR	**R251**	(FBh)	Page 0	Read/Write	Timer/Watchdog Control Register	67

10 SLICE TIMER

STH	**R240**	(F0h)	Page 0B	Read/Write	Counter High-Byte Register	71
STL	**R241**	(F1h)	Page 0B	Read/Write	Counter Low-Byte Register	71
STP	**R242**	(F2h)	Page 0B	Read/Write	Slice Timer Prescaler Register	71
STC	**R243**	(F3h)	Page 0B	Read/Write	Slice Timer Control Register	71

11 ON SCREEN DISPLAY

HDCS	**R244**	(F4h)	Page 2A	Read/Write	Horizontal Delay/Character Size Register	74
AR	**R245**	(F5h)	Page 2A	Read/Write	Active Range Register	75
SL	**R249**	(F9h)	Page 2A	Read-only	Scan Line Register	78
FB	**R250**	(FAh)	Page 2A	Read-only	Flag Bit Register	78
BCOL	**R251**	(FBh)	Page 2A	Read/Write	Border Color Register	78
EN	**R247**	(F7h)	Page 2A	Read/Write	Enable Register	79
EL	**R246**	(F6h)	Page 2A	Read/Write	Event Line Register	79
VD	**R248**	(F8h)	Page 2A	Read/Write	Vertical Delay Register	79

12 IFP A/D CONVERTER

DTR	**R240**	(F0h)	Page 3E	Read/Write	Channel i Data Register	84

13 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	– 0.3 to 7.0	V
V_I	Input Voltage	V_{SS} – 0.3 to V_{DD} +0.3	V
V_O	Output Voltage	V_{SS} – 0.3 to V_{DD} +0.3	V
T_{STG}	Storage Temperature	– 55 to + 150	°C
I_{INJ}	Pin Injection Current Digital Input	-5 to +5	mA
	Maximum Accumulated Pin Injection Current in the device	-50 to +50	mA

Note: Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability. All voltages are referenced to V_{SS}

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Value		Unit
		Min.	Max.	
T_A	Operating Temperature	0	70	°C
V_{DD}	Operating Supply Voltage	4.5	5.5	V
f_{OSCE}	External Oscillator Frequency		24	MHz
f_{OSCI}	Internal Clock Frequency		12	MHz

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5V ± 10% T_A = 0°C to + 70°C, unless otherwise specified)

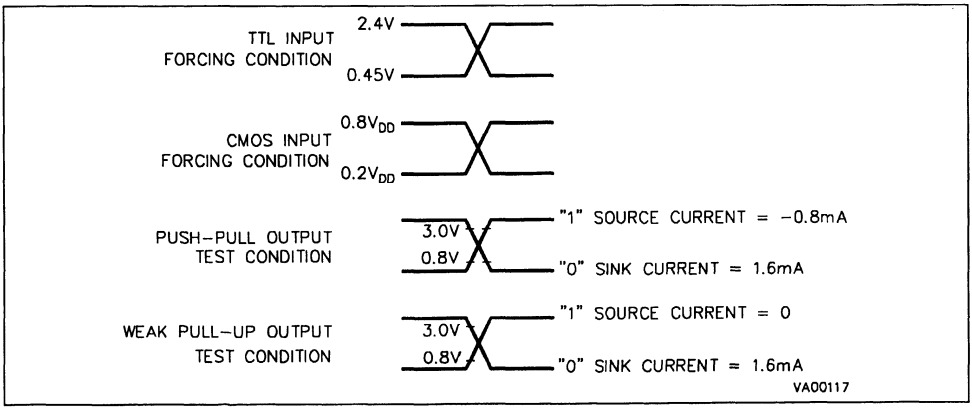
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IHCK}	Clock Input High Level	External Clock	0.7 V_{DD}			V
V_{ILCK}	Clock Input Low Level	External Clock			0.3 V_{DD}	V
V_{IH}	Input High Level	TTL	2.0			V
		CMOS	0.7 V_{DD}			V
V_{IL}	Input Low Level	TTL			0.8	V
		CMOS			0.3 V_{DD}	V
V_{IHRS}	$\overline{\text{RESET}}$ Input High Level		0.7 V_{DD}			V
V_{ILRS}	$\overline{\text{RESET}}$ Input Low Level				0.3 V_{DD}	V
V_{HYRS}	$\overline{\text{RESET}}$ Input Hysteresis		0.3			V
V_{OH}	Output High Level	Push Pull, Iload = – 4mA	V_{DD} – 0.8			V
V_{OL}	Output Low Level	Push Pull or Open Drain, Iload = 1.6mA			0.4	V

ELECTRICAL CHARACTERISTICS Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
I_{WPU}	Weak Pull-up Current	Bidirectional Weak Pull-up, V_{OL} = 0V	– 80	– 200	– 420	µA
I_{LKIO}	I/O Pin Input Leakage	Input/Tri-State, 0V < V_{IN} < V_{DD}	– 10		+ 10	µA
I_{LKRS}	$\overline{\text{Reset}}$ Pin Input Leakage	0V < V_{IN} < V_{DD}	– 30		+ 30	µA
I_{LKAD}	A/D Pin Input Leakage	Alternate Function, Open Drain, 0V < V_{IN} < V_{DD}	– 3		+ 3	µA
I_{LKOS}	OSCIN Pin Input Leakage	0V < V_{IN} < V_{DD}	– 10		+ 10	µA
I_{DD}	Run Mode Current no CPUCLK prescale, Clock divide by 2	24MHz, Note 1		40	70	mA
I_{DP2}	Run Mode Current Prescale by 2 Clock divide by 2	24MHz, Note 1		19	40	mA
I_{WFI}	WFI Mode Current no CPUCLK prescale, Clock divide by 2	24MHz, Note 1		11	18	mA
I_{HALT}	HALT Mode Current	24MHz, Note 1		50	100	µA

Note:
1. All I/O Ports are configured in Bidirectional Weak Pull-up Mode with no DC load, External Clock pin (OSCIN) is driven by square wave external clock. No peripheral working. External interface not active (Internal Program Execution).

AC TEST CONDITIONS



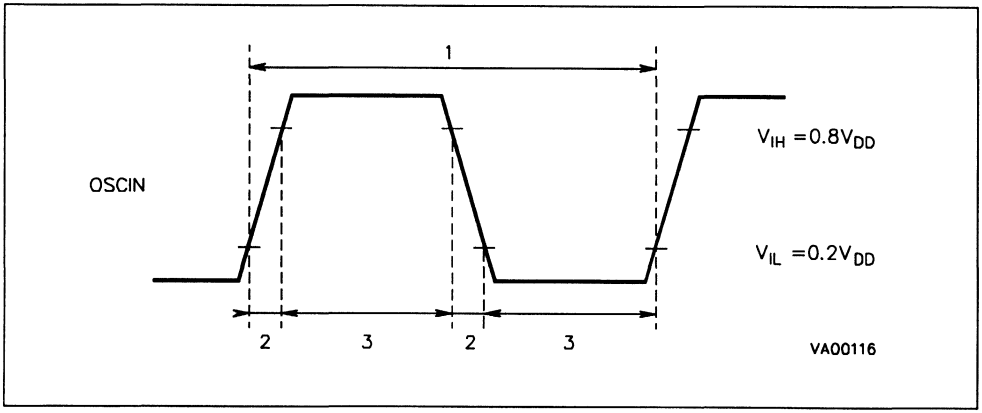
CLOCK TIMING TABLE

(V_{DD} = 5V ± 10%, T_A = 0°C to + 70°C, INTCLK = 12MHz, unless otherwise specified)

N°	Symbol	Parameter	Value		Unit	Note
			Min.	Max.		
1	TpC	OSCIN Clock Period	41.5		ns	1
			83		ns	2
2	TrC, TfC	OSCIN Rise and Fall Time		12	ns	
3	TwCL, TwCH	OSCIN Low and High Width	17	25	ns	1
			38		ns	2

Notes:
1. Clock divided by 2 internally (MODER.DIV2=1)
2. Clock not divided by 2 internally (MODER.DIV2=0)

CLOCK TIMING

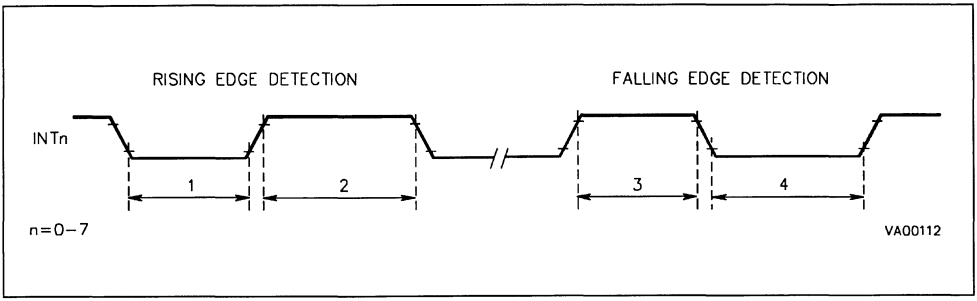


EXTERNAL INTERRUPT TIMING TABLE (V_{DD} = 5V ± 10%, T_A = 0°C to +70°C, Cload = 50pF, INTCLK = 12MHz, Push-pull output configuration, unless otherwise specified)

N°	Symbol	Parameter	Value (Note)				Unit
			OSCIN Divided By 2 Min.	OSCIN Not Divided By 2 Min.	Min.	Max.	
1	TwLR	Low Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95		ns
2	TwHR	High Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95		ns
3	TwHF	High Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95		ns
4	TwLF	Low Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95		ns

Note: The value left hand two columns show the formula used to calculate the timing minimum or maximum from the oscillator clock period, prescale value and number of wait cycles inserted.
The value right hand two columns show the timing minimum and maximum for an external clock at 24 MHz divided by 2, prescale value of zero and zero wait status.

EXTERNAL INTERRUPT TIMING

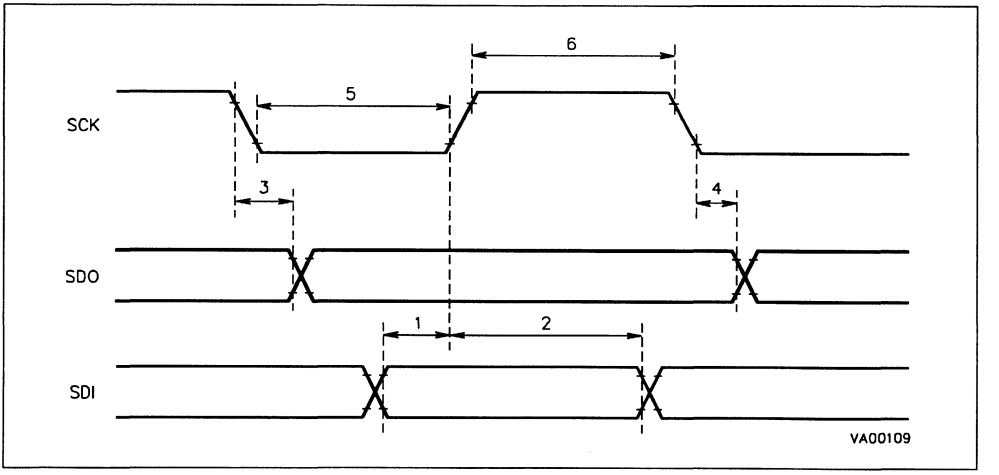


SPI TIMING TABLE (V_{DD} = 5V ± 10%, T_A = 0°C to +70°C, Cload = 50pF, INTCLK = 12MHz, Output Alternate Function set as Push-pull)

N°	Symbol	Parameter	Value		Unit
			Min.	Max.	
1	TsDI	Input Data Set-up Time	100		ns
2	ThDI (1)	Input Data Hold Time	1/2 TpC+100		ns
3	TdOV	SCK to Output Data Valid		100	ns
4	ThDO	Output Data Hold Time	-20		ns
5	TwSKL	SCK Low Pulse Width	300		ns
6	TwSKH	SCK High Pulse Width	300		ns

Note: TpC is the OSCIN Clock period.

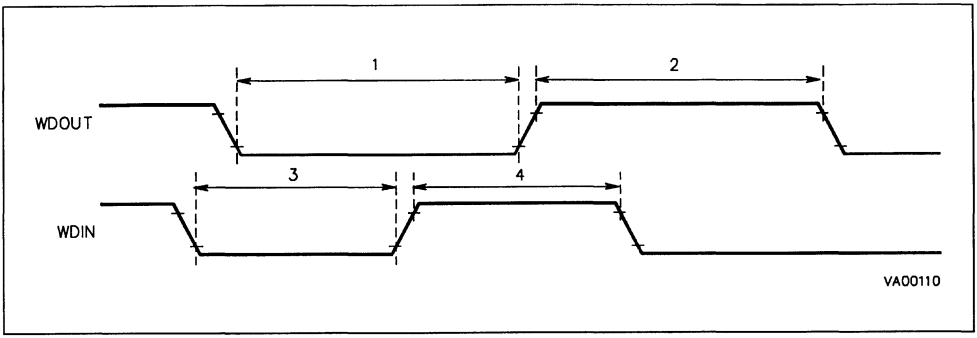
SPI TIMING



WATCHDOG TIMING TABLE(V_{DD} = 5V ± 10%, T_A = 0 °C to +70°C, Cload = 50pF, CPUCLK = 12MHz, Push-pull output configuration, unless otherwise specified)

N°	Symbol	Parameter	Values		Unit
			Min.	Max.	
1	TwWDOL	WDOUT Low Pulse Width	620		ns
2	TwWDOH	WDOUT High Pulse Width	620		ns
3	TwWDIL	WDIN High Pulse Width	350		ns
4	TwWDIH	WDIN Low Pulse Width	350		ns

WATCHDOG TIMING



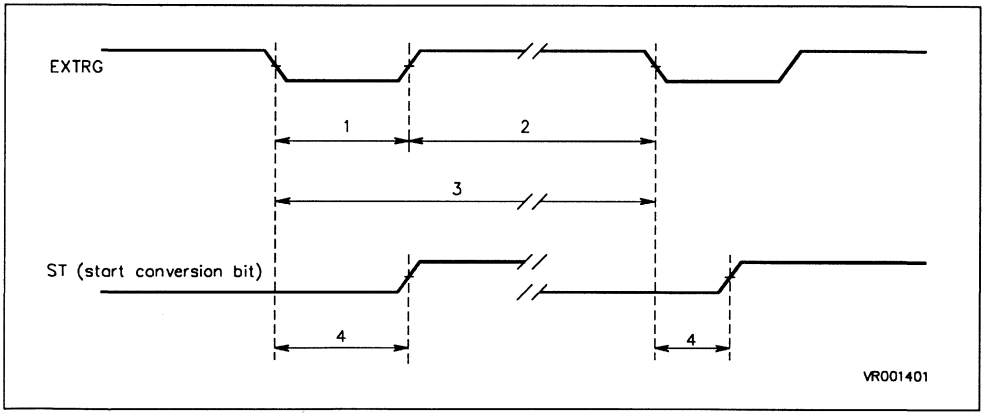
A/D CONVERTER

EXTERNAL TRIGGER TIMING (V_{DD} = 5V ± 10%, T_A = - 0°C to + 70°C, Cload = 50pF)

N°	Symbol	Parameter	Oscin divided by 2 (1)		Oscin not divided (1)		Value(2)		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
1	T_{LOW}	External Trigger pulse width	$2xT_{PC}$		T_{PC}		83		ns
2	T_{HIGH}	External Trigger pulse distance	$2xT_{PC}$		TPC		83		ns
3	T_{EXT}	Eternal trigger active edges distance	$138xT_{PC}$		68xTPC		4.5		µs
4	T_{STR}	Internal delay between EXTRG falling edge and first conversion start	T_{PC}	3xTPC	0.5xTPC	1.5xTPC	41.5	125	ns

Notes:
1. Variable clock (TPC=OSCIN clock period)
2. CPUCLK=12MHz

A/D External Trigger Timing



ANALOG SPECIFICATIONS (V_{CC} = 5V ± 10%, T_A = 0 to +70°C)

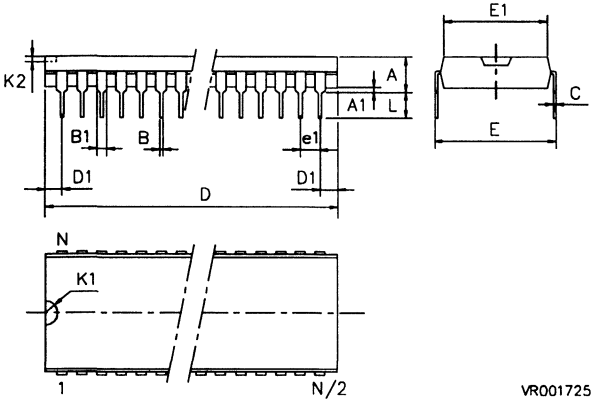
Parameter	Min.	Typical[1]	Max.	Units	Notes
Analog Input Range	V_{SS}		V_{CC}	V	
Conversion Time	5.75			µs	(2,3)
Sample Time	1.5			µs	(2)
Power-up Time	60			µs	
Resolution		8		bit	
Monotonicity	GUARANTEED				
No missing Codes					
Zero Input reading	00				
Full scale reading			FF	Hex	
Offset Error	TBD			LSBs[5]	
Gain Error	TBD			LSBs	
Diff. Non lin.	-1		+2	LSBs	
Int Non Lin.	-2		+2	LSBs	
Absolute Accuracy			+2	LSBs	
S/N			49	dB	
Input Resistance	8	12	15	KΩ	(4)
Hold capacitance Max.			15	pF	

Notes:
1. The values are expected at 25°C with V_{CC}= 5V
2. At 24MHz external clock
3. Including sample time
4. It must be intended as the internal series resistance before the sampling capacitor
5. "LSBs", as used here, has a value of V_{CC}/256

S/N ratio valued sampling a sinusoidal input and calculating its Fast Fourier Transform.

PACKAGE MECHANICAL DATA

42-Pin Plastic Shrink Dual-In-line Package, 600 Mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D	36.32		39.12	1.430		1.540
D1	–	–	–	–	–	–
E			18.54			0.730
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	42					

ORDERING INFORMATION

Sales Type	ROM SIZE	RAM SIZE	Temperature Range	Package
ST9293J7B1/XX	48K	768	0°C to + 70°C	PSDIP42
ST9293J5B1/XX	32K	640		PSDIP42
ST9293J3B1/XX	24K	512		PSDIP42
ST9293J1B1/XX	16K	256		PSDIP42

Note: "XX" is the ROM code identifier that is allocated by SGS-THOMSON after receipt of all required options and the related ROM file.

ST9293 OPTION LIST
Please copy this page (enlarge if possible) and complete ALL sections.
Send the form, with the ROM code image required, to your local SGS-THOMSON sales office.

Customer Company : [.....]

Company Address : [.....]

[.....]

Telephone : [.....]

FAX : [.....]

Contact : [.....] Telephone (Direct) : [.....]

Please confirm characteristics of device :

Device [ ] ST9293J1 (16K ROM, 256 RAM) [ ] ST9293J3 (24K ROM, 512 RAM)

[ ] ST9293J5 (32K ROM, 640 RAM) [ ] ST9293J7 (48K ROM, 768 RAM)

Package : PSDIP42
Temperature Range : 0 to 70°C

Special Marking [ ] No

[ ] Yes 14 characters [| | | | | | | | | | | | | |]

Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Please consult your local SGS-THOMSON sales office for other marking details if required.

Notes :

OSD Code (INTEL ® HEX format files on PCcompatible disk)
[ ] Separate EVEN/ODD byte Filename [..... .EV%]
Filename [..... .OD%]
[ ] Single interleaved file Filename [.....]

Code : [ ] ROM (27128, 27256)
[ ] INTEL ® HEX format file on PCcompatible disk Filename [.....]

Confirmation : [ ] Code checked with EPROM device in application

Yearly Quantity forecast : [.....] k units

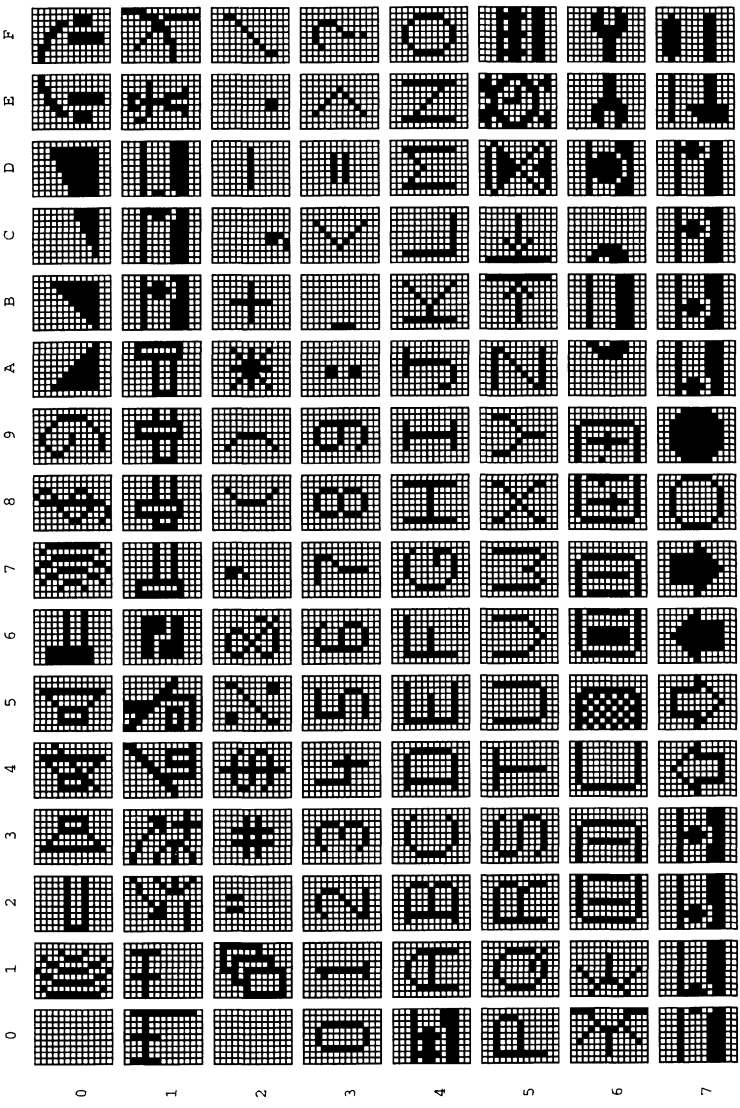
- for a period of : [.....] years

Preferred Production start dates : [.....] (YY/MM/DD)

Customer Signature :
Date :

APPENDIX 1

Sample OSD Character Set



VR001824

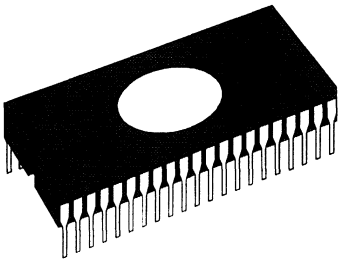
SGS-THOMSON MICROELECTRONICS

ST92E93
ST92T93

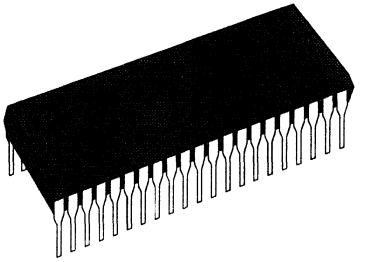
48K/32K EPROM AND OTP HCMOS MCUs WITH ON SCREEN DISPLAY AND CLOSED-CAPTION DATA SLICER

PRELIMINARY DATA

- Register oriented 8/16 bit CORE with RUN, WFI and HALT modes
- Minimum instruction cycle time: 500ns (12MHz internal)
- 48K/32K bytes of EPROM, 768/640 bytes of RAM, 224 general purpose registers available as RAM, accumulators or index registers (Register File)
- 42-lead Window Ceramic Shrink DIP package for ST92E93
- 42-lead Plastic Shrink DIP package for ST92T93
- Interrupt handler and Serial Peripheral Interface as standard features
- 31 fully programmable I/O pins
- 34 character x15 rows software programmable On Screen Display module with colour, italic, underline, flash, transparent and fringe attribute options
- 16 bit Timer with 8 bit Prescaler, able to be used as a Watchdog Timer
- 16-bit programmable Slice Timer with 8-bit prescaler
- 4 channel Analog to Digital Converter, with integral sample and hold, fast 5.75µs conversion time, 6-bit guaranteed resolution
- Rich Instruction Set and 14 Addressing modes
- Division-by-Zero trap generation
- Versatile Development tools, including assembler, linker, C-compiler, archiver, graphic oriented debugger and hardware emulators
- Real Time Operating System
- Compatible with ST9293 ROM device



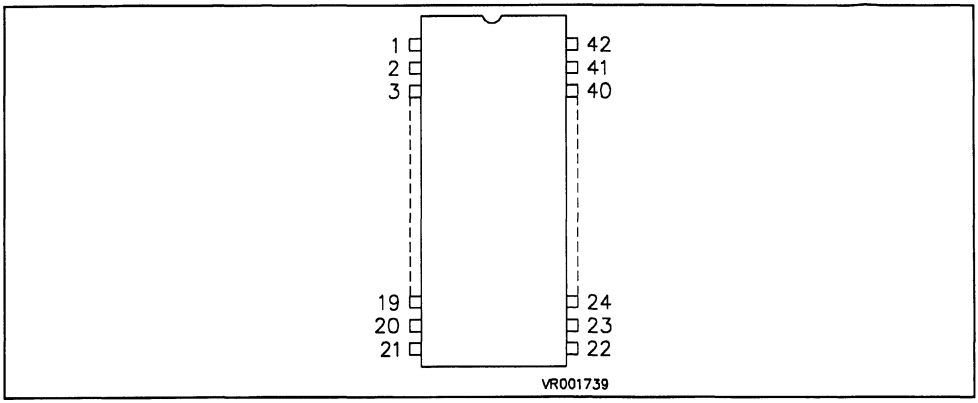
CSDIP42-W



PSDIP42

(Ordering Information at the end of the datasheet)

Figure 1. 42 Pin Shrink DIP Pinout



ST92E93J Pin Description

Pin	Pin name
1	P2.4/WDOUT/EXTRG (AD)
2	P2.5/SDO
3	P2.6/SCK/INT2
4	P2.7/SDI/SDO
5	P0.7
6	P0.6
7	P0.5
8	P0.4
9	P0.3
10	P0.2
11	P0.1
12	P0.0
13	V_{DD}
14	P5.0
15	P5.1
16	P5.2
17	P5.3/FB
18	P5.4/B
19	P5.5/G
20	P5.6/R
21	V_{SS}

Pin	Pin name
42	P2.3/INT3
41	P2.2/INT0
40	P2.1/INT7
39	P2.0/INT6
38	P3.7/NMI
37	P3.6/AIN4/WDIN
36	P3.5/INT4
35	P3.4-OD/SLIN
34	P3.3-OD/SLOUT
33	$\overline{RESET}/V_{PP}$
32	OSCIN
31	V_{SS2}
30	OSCOUT
29	P4.7/AIN7
28	P4.6/AIN6
27	P4.5/AIN5
26	VSYNC
25	HSYNC
24	AV_{DD}
23	PLLR
22	PLLF

14.1 GENERAL DESCRIPTION

The ST92E93 is an EPROM member in windowed ceramic (E) and plastic OTP (T) packages of the ST9 family of microcontrollers, completely developed and produced by SGS-THOMSON Microelectronics using a n-well proprietary HCMOS process.

The EPROM parts are fully compatible with their ROM versions and this datasheet will thus provide only information specific to the EPROM based devices.

THE READER IS ASKED TO REFER TO THE DATASHEET OF THE ST9293 ROM-BASED DEVICE FOR FURTHER DETAILS.

The EPROM ST92E93 may be used for the prototyping and pre-production phases of development.

The nucleus of the ST92E93 is the advanced Core which includes the Central Processing Unit (CPU), the Register File, a 16-bit Timer/Watchdog with 8-bit Prescaler, a Serial Peripheral Interface supporting S-bus, I^2C-bus and IM-bus Interface, plus two 8-bit I/O ports. The Core has independent memory and register buses allowing a high degree of pipelining to add to the efficiency of the code execution speed of the extensive instruction set.The powerful I/O capabilities demanded by microcontroller applications are fulfilled by the ST92E93 with up to 31 I/O lines dedicated to digital Input/Output.

These lines are grouped into up to six I/O Ports and can be configured on a bit basis under software-control to provide timing, status signals, timer inputs and outputs, analog inputs, external interrupts and serial or parallel I/O.

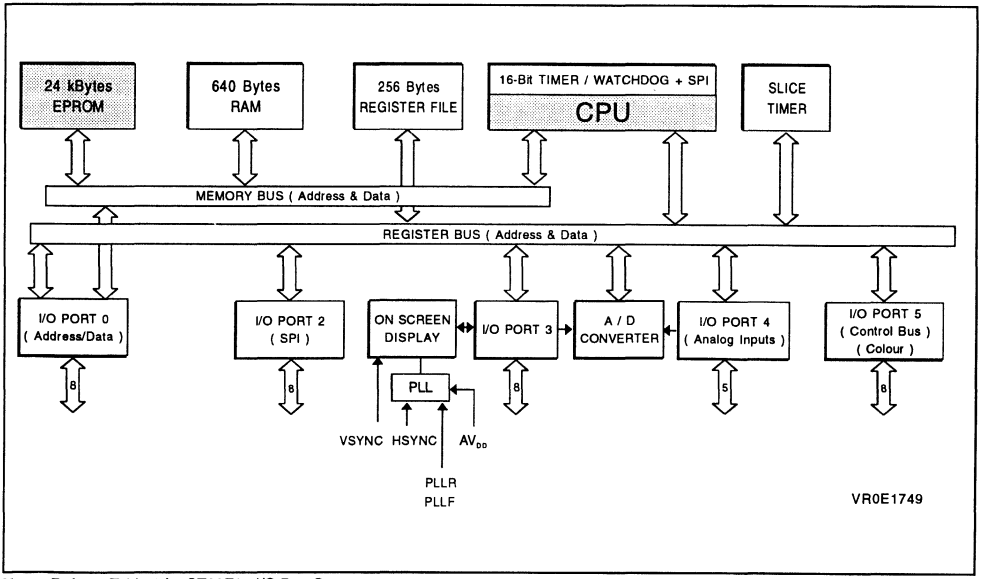
Three basic memory spaces are available to support this wide range of configurations: Program Memory, Data Memory and the Register File, which includes the control and status registers of the on-chip peripherals.

The 16-bit Slice Timer with an 8-bit Prescaler and 6 operating modes allows simple use for waveformgeneration and measurement, PWM functions and many other system timing functions.

The human interface is provided by the On Screen Display module, this can produce up to 8 lines of of up to 34 characters from a ROM defined 128 character set. The 9x13 character can be modified by 4 different pixel sizes, with character rounding, and formed into words with colour and format attributes.

In addition there is a 4 channel Analog to Digital Converter with integral sample and hold, fast 5.5µs conversion time and 6-bit guaranteed resolution.

Figure 2.ST92E93 Block Diagram



Note : Refer to Table 1 for ST92E94 I/O Port Summary

14.2PIN DESCRIPTION

VSYNC. *Vertical Synch.* Vertical video synchronisation input to OSD. Positive or negative polarity.

HSYNC. *Horizontal Synch.* Horizontal video synchronisation input to OSD. Positive or negative polarity.

PLLF. *PLL Filter input.* Filter input for the OSD for PLL feed-back.

PLLR. *PLL Resistor connection pin.* Resistor connection to select the PLL gain adjust.

$\overline{\text{RESET}}$/V_{PP}. *Reset (input, active low) or V_{PP} (input).* The ST9 is initialised by the Reset signal. With the deactivation of $\overline{\text{RESET}}$, program execution begins from the Program memory location pointed to by the vector contained in program memory locations 00h and 01h. In the EPROM programming Mode, this pin acts as the programming voltage input V_{PP}.

OSCIN, OSCOUT. *Oscillator (input and output).* These pins connect a parallel-resonant crystal (24MHz maximum), or an external source to the on-chip clock oscillator and buffer. OSCIN is the input of the oscillator inverter and internal clock generator; OSCOUT is the output of the oscillator inverter.

AV_{DD}. Analog V_{DD} of PLL. This pin must be tied to V_{DD} externally to the ST92E93.

V_{DD}. Main Power Supply Voltage (5V±10%)

V_{SS}, V_{SS2}. Digital Circuit Ground, these pins must be connected together externally to the ST92E93.

P0.0-P0.7, P2.0-P2.7, P3.3-P3.7, P4.5-P4.7, P5.0-P5.6 (J suffix) *I/O Port Lines (Input/Output, TTL or CMOS compatible).* 31 lines grouped into I/O ports, bit programmable under program control as general purpose I/O or as Alternate functions (see next section).
P3.3 and P3.4 are true 12V open drain outputs when set in output mode.

I/O Port Alternate Functions.

Each pin of the I/O ports of the ST92E93 may assume software programmable Alternative Functions as shown in the Pin Configuration Drawings. Table 1 shows the Functions allocated to each I/O Port pin.

Table 1. ST92E93 I/O Port Alternate Function Summary

I/O PORT	Name	Function	Alternate Function	Pin Assignment
Port.bit				92E93J
P0.0		I/O		12
P0.1		I/O		11
P0.2		I/O		10
P0.3		I/O		9
P0.4		I/O		8
P0.5		I/O		7
P0.6		I/O		6
P0.7		I/O		5
P2.0	INT6	I	External Interrupt 6	39
P2.1	INT7	I	External Interrupt 7	40
P2.2	INT0	I	External Interrupt 0	41
P2.3	INT3	I	External Interrupt 3	42
P2.4	WDOUT	O	T/WD Output	1
P2.4	EXTRG	I	External A/D Trigger	1

PIN DESCRIPTION (Continued)

Table 2. ST92E93 I/O Port Alternate Function Summary

I/O PORT	Name	Function	Alternate Function	Pin Assignment
Port.bit				92E93J
P2.5	SDO	O	MSPI Serial Data Output	2
P2.6	INT2	I	External Interrupt 2	3
P2.6	SCK	O	SPI Serial Clock	3
P2.7	SDO	O	SPI Serial Data Output	4
P2.7	SDI	I	SPI Serial Data Input	4
P3.3		O	(12V Open Drain Output)	34
P3.3	SLOUT	O	Slice Timer Output	34
P3.4		O	(12V Open Drain Output)	35
P3.4	SLIN	I	Slice Timer Input	35
P3.5	INT4	I	Schmitt Triggered Input Only	36
P3.6	WDIN	I	T/WD Input	37
P3.6	AIN4	I	A/D Analog Input 4	37
P3.7	NMI	I	Non-Maskable Interrupt	38
P4.5	AIN5	I	A/D Analog Input 5	27
P4.6	AIN6	I	A/D Analog Input 6	28
P4.7	AIN7	I	A/D Analog Input 7	29
P5.0		I/O		14
P5.1		I/O		15
P5.2		I/O		16
P5.3	FB	O	Fast Blanking OSD output	17
P5.4	B	O	Blue Video Colour OSD output	18
P5.5	G	O	Green Video Colour OSD output	19
P5.6	R	O	Red Video Colour OSD output	20

14.3 MEMORY

The memory of the ST92E93 is functionally divided into two areas, the Register File and Memory. The Memory may optionally be divided into two spaces, each having a maximum of 65,536 bytes. The two memory spaces are separated by function, one space for Program code, the other for Data. The ST92E93 48K/32K bytes of on-chip ROM memory is selected in the PROGRAM space while the ST92T93 OTP version has the top 64 bytes of the EPROM reserved by SGS-THOMSON for testing purposes. The on-chip RAM is mapped into both Program and Data space (the P/D bit is not used for the decoding).

The on-chip general purpose (GP) Registers may also be used as RAM memory for minimum chip count systems.

14.4 EPROM PROGRAMMING

The 24576 bytes of EPROM memory of the ST92E93 (24512 for the ST92T93) may be programmed by using the EPROM Programming Boards (EPB) available from SGS-THOMSON.

14.4.1 EPROM Erasing

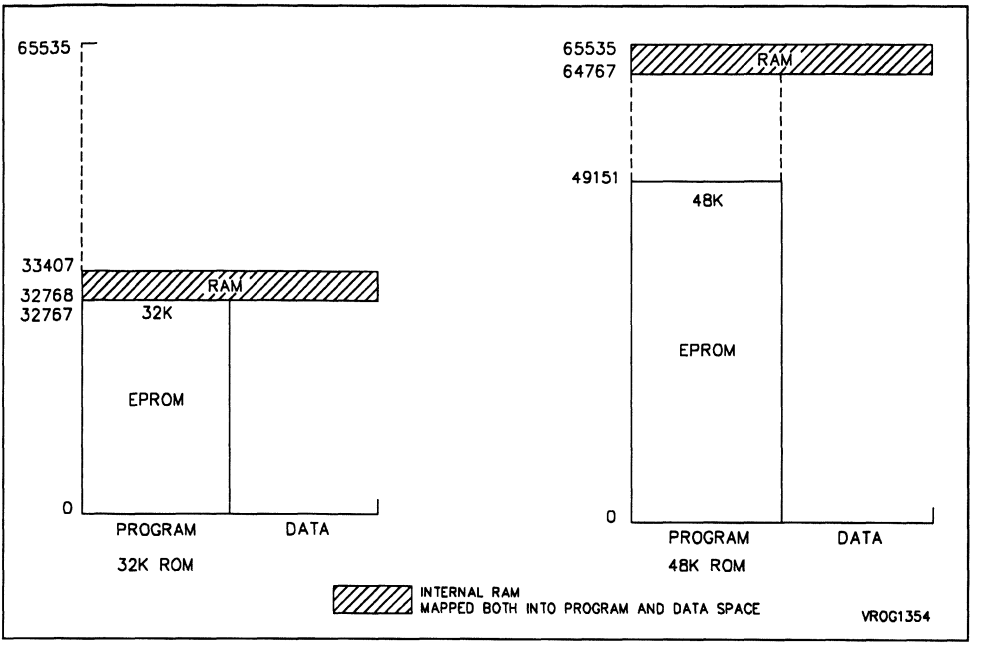
The EPROM of the windowed package of the ST92E93 may be erased by exposure to Ultra-Violet light.
The erasure characteristic of the ST92E93 is such that erasure begins when the memory is exposed to light with a wave length shorter than approximately 4000Å. It should be noted that sunlight and some types of fluorescent lamps have wavelengths in the range 3000-4000Å. It is thus recommended that the window of the ST92E93 packages be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.

The recommended erasure procedure of the EPROM is the exposure to short wave ultraviolet light which have a wave-length 2537Å. The integrated dose (i.e. U.V. intensity x exposure time) for erasure should be a minimum of 15W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm^2 power rating. The ST92E93 should be placed within 2.5cm (1Inch) of the lamp tubes during erasure.

Table 3. ROM and RAM Address Configuration

Device Suffix		EPROM Size (bytes)	EPROM Addresses		RAM Size (bytes)	RAM Addresses	
J7	EPROM	48K	0 - 49151	dec	768	64767 - 65536	dec
			0000 - BFFF	hex			
	OTP		0 - 49087	dec		F000 - FFFF	hex
			0000 - BFBF	hex			
J6	EPROM	32K	0 - 32767	dec	640	32768 - 33407	dec
			0000 - 7FFF	hex			
	OTP		0 - 32703	dec		8000 - 827F	hex
			0000 - 7FBF	hex			

Figure 3. Memory Map



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	V_{SS} – 0.3 to 7.0	V
AV_{DD}	Analog Supply Voltage	$AV_{DD} \geq V_{SS}$ - 0.3 to 7.0 V_{DD} - 0.3 ≤ AV_{DD} ≤ V_{DD} + 0.3	V
V_I	Input Voltage	V_{SS} – 0.3 to V_{DD} + 0.3	V
V_O	Output Voltage	V_{SS} – 0.3 to V_{DD} + 0.3	V
T_{STG}	Storage Temperature	– 55 to + 150	°C

Note: Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Value		Unit
		Min.	Max.	
T_A	Operating Temperature	0	70	°C
V_{DD}	Operating Supply Voltage	4.5	5.5	V
f_{OSCE}	External Oscillator Frequency		24	MHz
f_{OSCI}	Internal Oscillator Frequency		12	MHz

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5V ± 10% T_A = 0°C to + 70°C, unless otherwise specified)

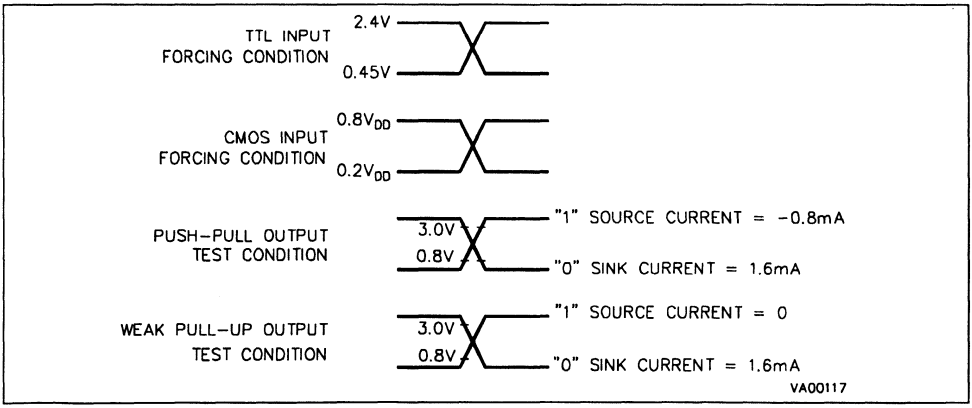
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IHCK}	Clock Input High Level	External Clock	0.7 V_{DD}			V
V_{ILCK}	Clock Input Low Level	External Clock			0.3 V_{DD}	V
V_{IH}	Input High Level	TTL	2.0			V
		CMOS	0.7 V_{DD}			V
V_{IL}	Input Low Level	TTL			0.8	V
		CMOS			0.3 V_{DD}	V
V_{IHRS}	$\overline{\text{RESET}}$ Input High Level		0.7 V_{DD}			V
V_{ILRS}	$\overline{\text{RESET}}$ Input Low Level				0.3 V_{DD}	V
V_{HYRS}	$\overline{\text{RESET}}$ Input Hysteresis		0.3			V
V_{IHY}	P2.0, P2.1 Input Hysteresis		0.9		1.5	V
V_{HYHV}	HSYNC/VSYNC Hysteresis		0.5			V

ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{OH}	Output High Level	Push Pull, Iload = 4mA	V_{DD} – 0.8			V
V_{OL}	Output Low Level	Push Pull or Open Drain, Iload = 2mA			0.4	V
I_{WPU}	Weak Pull-up Current	Bidirectional Weak Pull-up, V_{OL} = 0V	– 80	– 200	– 420	µA
I_{OH}	High Voltage Open Drain Output Leakage Current	V_{OH} =12V			+ 30	
I_{LKIO}	I/O Pin Input Leakage	Input/Tri-State, 0V < V_{IN} < V_{DD}	– 10		+ 10	µA
I_{LKRS}	$\overline{\text{RESET}}$ Pin Input Leakage	0V < V_{IN} < V_{DD}	– 30		+ 30	µA
I_{LKAD}	A/D Pin Input Leakage	Alternate Function, Open Drain, 0V < V_{IN} < V_{DD}	– 3		+ 3	µA
I_{LKOS}	OSCIN Pin Input Leakage	0V < V_{IN} < V_{DD}	– 10		+ 10	µA

Note:
1. All I/O Ports are configured in Bidirectional Weak Pull-up Mode with no DC load, External Clock pin (OSCIN) is driven by square wave external clock. No peripheral working (Internal Program Execution).

AC TEST CONDITIONS



AC ELECTRICAL CHARACTERISTICS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
I_{DD}	Run Mode Current no CPUCLK prescale, CLOCK divide by 2	24MHz, Note 1		40	70	mA
I_{DP2}	Run Mode Current Prescale by 2 CLOCK divide by 2	24MHz, Note 1		19	40	mA
I_{WFI}	WFI Mode Current no CPUCLK prescale, CLOCK divide by 2	24MHz, Note 1		15	20	mA
I_{HALT}	HALT Mode Current	24MHz, Note 1		50	100	µA
I_{LPR}	Low Power Reset Current, Note 2			TBD	TBD	µA

Notes:
1. All I/O Ports are configured in Bidirectional Weak Pull-up Mode with no DC load, External Clock pin (OSCIN) is driven by square wave external clock. No peripheral working.
2. All I/O Port connected to V_{DD} by 10kΩPull-up, External Clock pin (OSCIN) Stopped.

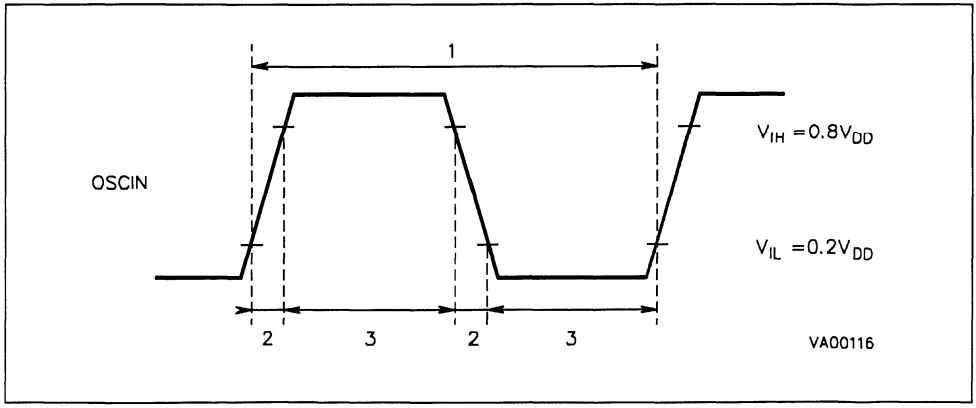
CLOCK TIMING TABLE

(V_{DD} = 5V ± 10%, T_A = − 40°C to + 85°C, INTCLK = 12MHz, unless otherwise specified)

N°	Symbol	Parameter	Value		Unit	Note
			Min.	Max.		
1	TpC	OSCIN Clock Period	41.5		ns	1
			83		ns	2
2	TrC, TfC	OSCIN Rise and Fall Time		12	ns	
3	TwCL, TwCH	OSCIN Low and High Width	17	25	ns	1
			38		ns	2

Notes:
1. Clock divided by 2 internally (MODER.DIV2=1)
2. Clock not divided by 2 internally (MODER.DIV2=0)

CLOCK TIMING

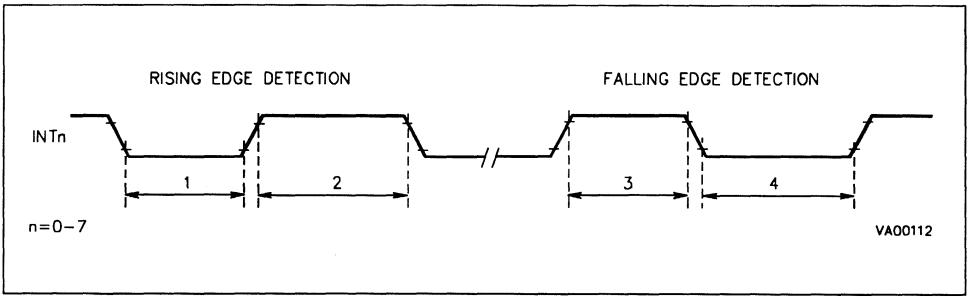


EXTERNAL INTERRUPT TIMING TABLE (V_{DD} = 5V ± 10%, T_A = 0°C to 70°C, Cload = 50pF, INTCLK = 12MHz, Push-pull output configuration, unless otherwise specified)

N°	Symbol	Parameter	Value (Note)				Unit
			OSCIN Divided By 2 Min.	OSCIN Not Divided By 2 Min.	Min.	Max.	
1	TwLR	Low Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95		ns
2	TwHR	High Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95		ns
3	TwHF	High Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95		ns
4	TwLF	Low Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95		ns

Note: The value left hand two columns show the formula used to calculate the timing minimum or maximum from the oscillator clock period, prescale value and number of wait cycles inserted.
The value right hand two columns show the timing minimum and maximum for an external clock at 24 MHz divided by 2, prescale value of zero and zero wait status.

EXTERNAL INTERRUPT TIMING

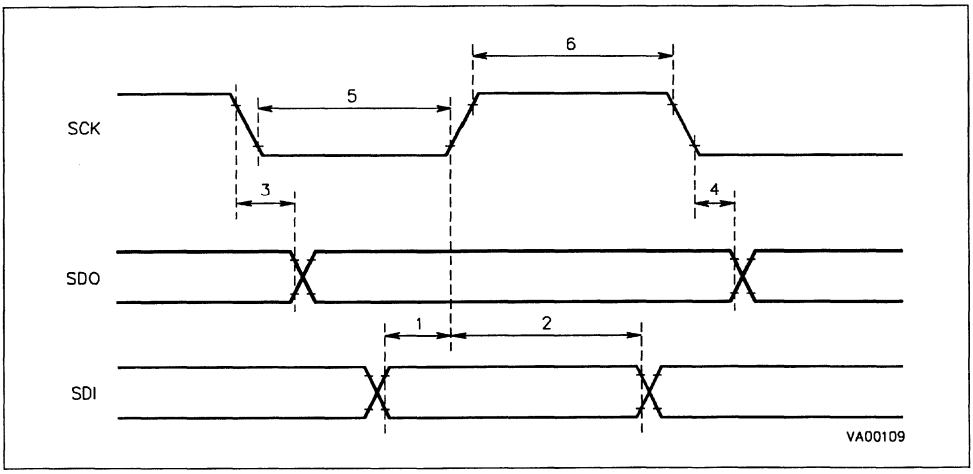


SPI TIMING TABLE (V_{DD} = 5V ± 10%, T_A = 0°C to 70°C, Cload = 50pF, INTCLK = 12MHz, Output Alternate Function set as Push-pull)

N°	Symbol	Parameter	Value		Unit
			Min.	Max.	
1	TsDI	Input Data Set-up Time	100		ns
2	ThDI (1)	Input Data Hold Time	1/2 TpC+100		ns
3	TdOV	SCK to Output Data Valid		100	ns
4	ThDO	Output Data Hold Time	-20		ns
5	TwSKL	SCK Low Pulse Width	300		ns
6	TwSKH	SCK High Pulse Width	300		ns

Note: TpC is the OSCIN Clock period.

SPI TIMING



ANALOG SPECIFICATIONS (V_{DD} = 5V ± 10%, T_A = +25°C)

Parameter	Min.	Typical[1]	Max.	Units	Notes
Analog Input Range	V_{SS}		V_{CC}	V	
Conversion Time	5.75			µs	(2,3)
Sample Time	1.5			µs	(2)
Power-up Time	60			µs	
Resolution		8		bit	
Monotonicity	GUARANTEED				
No missing Codes					
Zero Input reading	00				
Full scale reading			FF	Hex	
Offset Error	TBD			LSBs[5]	
Gain Error	TBD			LSBs	
Diff. Non lin.	-1		+2	LSBs	
Int Non Lin.	-2		+2	LSBs	
Absolute Accuracy			+2	LSBs	
S/N			49	dB	
Input Resistance	8	12	15	KΩ	(4)
Hold capacitance Max.			15	pF	

Notes:
1. The values are expected at 25°C with V_{CC}= 5V
2. At 24MHz external clock
3. Including sample time
4. It must be intended as the internal series resistance before the sampling capacitor
5. "LSBs", as used here, has a value of $V_{CC}/256$

S/N ratio valued sampling a sinusoidal input and calculating its Fast Fourier Transform.

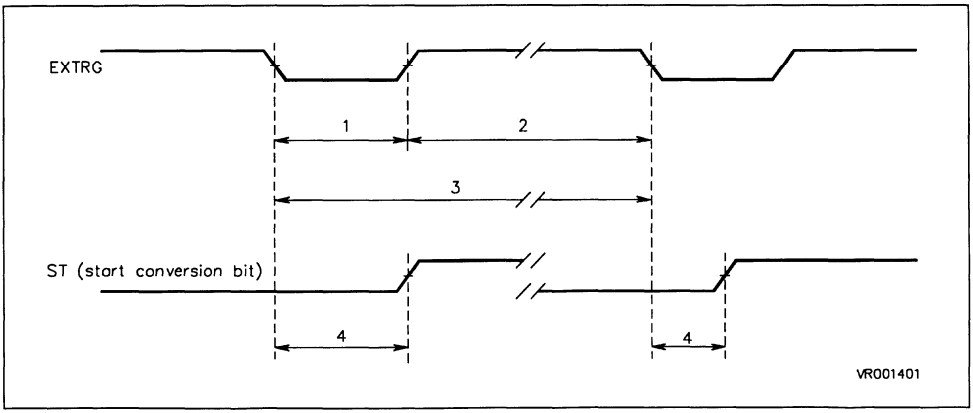
A/D CONVERTER

EXTERNAL TRIGGER TIMING (V_{DD} = 5V ± 10%, T_A = - 0°C to + 70°C, Cload = 50pF)

N°	Symbol	Parameter	Oscin divided by 2 (1)		Oscin not divided (1)		Value(2)		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
1	T_{LOW}	External Trigger pulse width	$2xT_{PC}$		T_{PC}		83		ns
2	T_{HIGH}	External Trigger pulse distance	$2xT_{PC}$		TPC		83		ns
3	T_{EXT}	External trigger active edges distance (fast mode)	$141xT_{PC}$		70.5xTPC		5.87		µs
4	T_{STR}	Internal delay between EXTRG falling edge and first conversion start	T_{PC}	3xTPC	$0.5xT_{PC}$	$1.5xT_{PC}$	41.5	125	ns

Notes:
1. Variable clock (TPC=OSCIN clock period)
2. CPUCLK=12MHz

A/D External Trigger Timing



ST92E93 STANDARD OPTION LIST
Please copy this page (enlarge if possible) and complete ALL sections.
Send the form, with the ROM code image required, to your local SGS-THOMSON sales office.

Customer Company : [.....]
Company Address : [.....]
[.....]
Telephone : [.....]
FAX : [.....]
Contact : [.....] Telephone (Direct) : [.....]

Please confirm characteristics of device :
Device [ ] ST92E93J7 (48K ROM, 768 RAM) [ ] ST92E93J5 (32K ROM, 640 RAM)
[ ] ST92T93J7 (48K ROM, 768 RAM) [ ] ST92T93J5 (32K ROM, 640 RAM)

Packages : PSDIP42, CSDIP42-W
Temperature Range : 0 to +70°C

Notes :

OSD Code (INTEL ® HEX format files on PC compatible disk)
[ ] Separate EVEN/ODD byte Filename [..... .EV%]
Filename [..... .OD%]
[ ] Single interleaved file Filename [.....]

Confirmation : [ ] Code checked with EPROM device in application

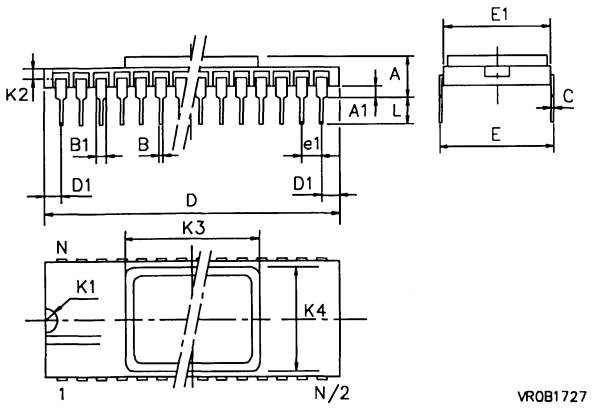
Yearly Quantity forecast : [.....] k units
- for a period of : [.....] years

Preferred Production start dates : [.....] (YY/MM/DD)

Customer Signature :
Date :

PACKAGE MECHANICAL DATA

42-Pin Ceramic Shrink Dual-In-line Package, 600 Mil Width with Window



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D	36.32		39.12	1.430		1.540
D1	–	–	–	–	–	–
E	15.2 BSD			0.600 BSD		
E1		13.72				0.540
K1		-			-	
K2		-			-	
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	42					

14.5 ORDERING INFORMATION

Sales Type	EPROM Size (bytes)	RAM Size (bytes)	Temperature Range	Package
ST92E93J7F1/XX	48K	768	0°C to+ 70°C	CSDIP42-W
ST92T93J7B1/XX	48K(1)	768		PSDIP42
ST92E93J5F1/XX	32K	640		CSDIP42-W
ST92T93J5B1/XX	32K(1)	640		PSDIP42

Notes: "XX" is the OSD ROM code identifier allocated by SGS-THOMSON after receipt of the related ROM file.
(1) The top 64 bytes of the EPROM Memory is reserved by SGS-THOMSON.

GENERAL INDEX

Pages

Pages

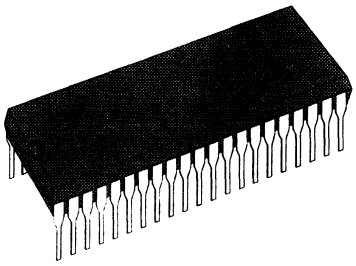
SGS-THOMSON MICROELECTRONICS

ST9294

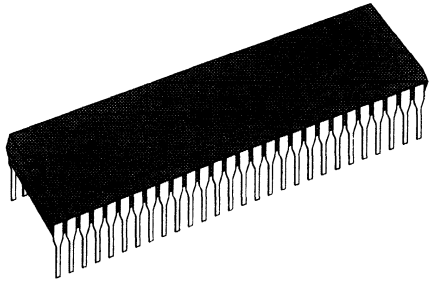
24K ROM HCMOS MCU WITH ON SCREEN DISPLAY AND CLOSED-CAPTION DATA SLICER

PRELIMINARY DATA

- Register oriented 8/16 bit CORE with RUN, WFI and HALT modes
- Minimum instruction cycle time: 500ns (12MHz internal)
- 12 to 32K bytes of ROM, 384/640 bytes of RAM, 224 general purpose registers available as RAM, accumulators or index registers (Register File)
- 42-lead Shrink DIP package or 56-lead Shrink DIP package
- Interrupt handler and Serial Peripheral Interface as standard features
- 31 (42 pin package) / 42 (56 pin package) fully programmable I/O pins
- 34 character x15 rows software programmable On Screen Display module with colour, italic, underline, flash, transparent and fringe attribute options
- Digital Data Slicer extracting closed caption data from video
- 8 8-bit PWM D/A outputs with repetition frequency 2 to 32kHz and 12V Open Drain Capability
- 16 bit Timer with 8 bit Prescaler, able to be used as a Watchdog Timer
- 16-bit programmable Slice Timer with 8-bit prescaler
- 3 channel Analog to Digital Converter, with integral sample and hold, fast 5.75µs conversion time, 6-bit guaranteed resolution
- Rich Instruction Set and 14 Addressing modes
- Division-by-Zero trap generation
- Versatile Development tools, including assembler, linker, C-compiler, archiver, graphic oriented debugger and hardware emulators
- Real Time Operating System
- Windowed EPROM parts available for prototyping and pre-production development phases



PSDIP42



PSDIP56

(Ordering Information at the end of the Datasheet)

Figure 1a. 42 Pin Shrink DIP Pinout

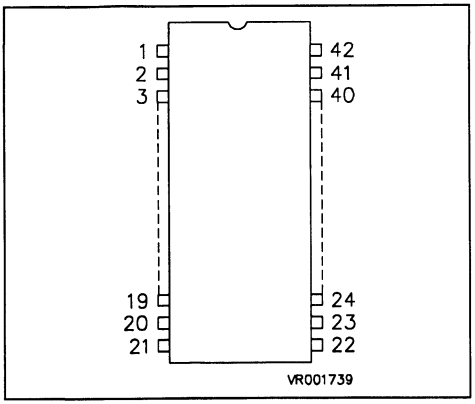
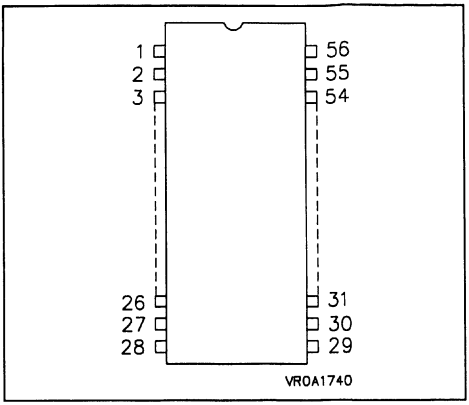


Figure 1b. 56 Pin Shrink DIP Pinout



ST9294J Pin Description

Pin	Pin name	Pin	Pin name
1	P2.0/INT7	42	P2.1/INT5/AIN1
2	$\overline{\text{RESET}}$	41	P2.2/INT0/AIN2
3	P0.7	40	P2.3/INT6/SDO
4	P0.6	39	P2.4/NMI
5	P0.5	38	P2.5/AIN3
6	P0.4	37	OSCIN
7	P0.3	36	OSCOUT
8	P0.2	35	P4.7/PWM7/ EXTRG (AD)
9	P0.1	34	P4.6/PWM6
10	P0.0	33	P4.5/PWM5
11	CCVideo	32	P4.4/PWM4
12	P3.6	31	P4.3/PWM3
13	P3.5	30	P4.2/PWM2
14	P3.4	29	P4.1/PWM1
15	P3.3/B	28	P4.0/PWM0
16	P3.2/G	27	VSYNC
17	P3.1/R	26	HSYNC
18	P3.0/FB	25	AV_{DD}
19	P5.1/SDIO	24	PLLR
20	P5.0/SCK/INT2	23	PLLF
21	V_{DD}	22	V_{SS}

ST9294N Pin Description

Pin	Pin name	Pin	Pin name
1	P2.1/INT5/AIN1	56	P2.2/INT0/AIN2
2	P2.0/INT7	55	P2.3/INT6/SDO
3	$\overline{\text{RESET}}$	54	P2.4/NMI
4	P0.7	53	P2.5/AIN3
5	P0.6	52	P1.0
6	P0.5	51	P1.1
7	NC	50	P1.2
8	P0.4	49	P1.3
9	P0.3	48	P1.4
10	P0.2	47	P1.5
11	P0.1	46	P1.6
12	P0.0	45	P1.7
13	NC	44	OSCIN
14	V_{DD}	43	OSCOUT
15	CCVideo	42	P4.7/PWM7/ EXTRG (AD)
16	P3.7	41	P4.6/PWM6
17	P3.6	40	P4.5/PWM5
18	P3.5	39	P4.4/PWM4
19	P3.4	38	P4.3/PWM3
20	P3.3/B	37	P4.2/PWM2
21	P3.2/G	36	P4.1/PWM1
22	P3.1/R	35	P4.0/PWM0
23	P3.0/FB	34	VSYNC
24	P5.3	33	HSYNC
25	P5.2	32	AV_{DD}
26	P5.1/SDIO	31	PLLR
27	P5.0/SCK/INT2	30	PLLF
28	V_{DD}	29	V_{SS}

1.1 GENERAL DESCRIPTION

The ST9294 is a ROM member of the ST9 family of microcontrollers, completely developed and produced by SGS-THOMSON Microelectronics using a proprietary n-well HCMOS process.

The ROM parts are fully compatible with their EPROM versions, which may be used for the prototyping and pre-production phases of development.

The nucleus of the ST9294 is the advanced Core which includes the Central Processing Unit (CPU), the Register File, a 16-bit Timer/Watchdog with 8-bit Prescaler, a Serial Peripheral Interface supporting S-bus, I^2C-bus and IM-bus Interface, plus two 8-bit I/O ports. The Core has independent memory and register buses allowing a high degree of pipelining to add to the efficiency of the code execution speed of the extensive instruction set.

The powerful I/O capabilities demanded by microcontroller applications are fulfilled by the ST9294 with up to 31/42 I/O lines dedicated to digital Input/Output. These lines are grouped into up to six I/O Ports and can be configured on a bit basis under software control to provide timing, status signals, timer inputs and outputs, analog inputs, external interrupts and serial or parallel I/O.

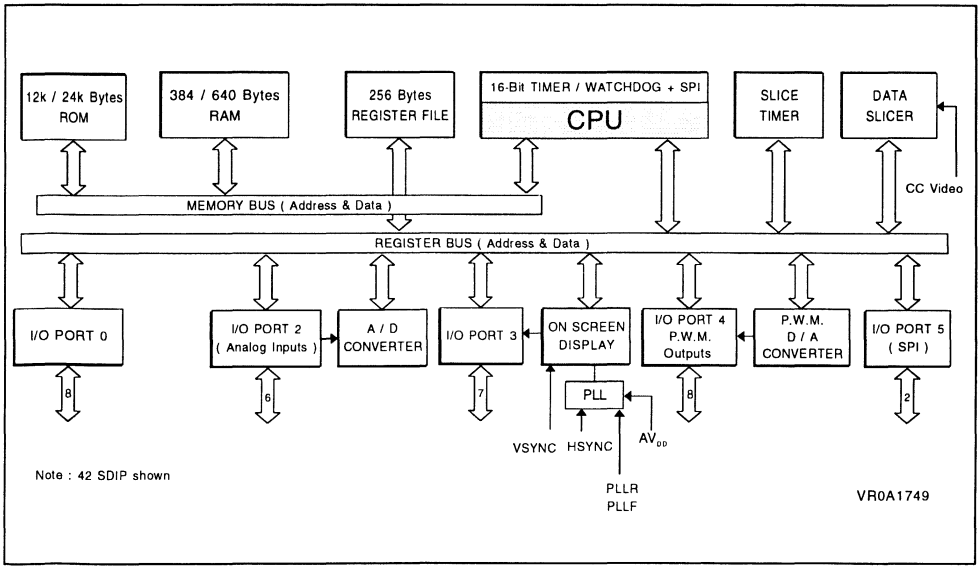
Three basic memory spaces are available to support this wide range of configurations: Program Memory, Data Memory and the Register File, which includes the control and status registers of the on-chip peripherals.

The 16-bit Slice Timer with an 8-bit Prescaler.

The human interface is provided by the On Screen Display module, this can produce up to 15 lines of up to 34 characters from a ROM defined 128 character set. The 9x13 character can be modified by 4 different pixel sizes, with character rounding, and formed into words with colour and format attributes.

Closed Caption control for the display of information transmitted through the video input is facilitated with the Data Slicer. This module has manual and automatic Slicing levels for both Sync and Data and allows the user to select the video line containing the data relative to the Vertical synchronisation pulse.

Figure 1-2. ST9294 Block Diagram



GENERAL DESCRIPTION (Continued)

Control of TV settings is able to be made with up to eight 8-bit PWM outputs, with a frequency maximum of 23,437Hz at 8-bit resolution (INTCLK = 12MHz). Low resolutions with higher frequency operation can be programmed.

In addition there is a 3 channel Analog to Digital Converter with integral sample and hold, fast 5.75µs conversion time and 6-bit guaranteed resolution.

1.2 PIN DESCRIPTION

VSYNC. *Vertical Synch.* Vertical video synchronisation input to OSD. Positive or negative polarity.

HSYNC. *Horizontal Synch.* Horizontal video synchronisation input to OSD. Positive or negative polarity.

CCVideo. *Composite Video Input.* Input to Data Slicer for Closed Caption extraction, 1V± 6dB or 2V± 3dB.

PLLF. *PLL Filter input.* Filter input for the OSD for PLL feed-back.

PLLR. *PLL Resistor connection pin.* For resistor connection to select the PLL gain adjust.

$\overline{\text{RESET}}$. *Reset (input, active low).* The ST9 is initialised by the Reset signal. With the deactivation of $\overline{\text{RESET}}$, program execution begins from the Program memory location pointed to by the vector contained in program memory locations 00h and 01h.

OSCIN, OSCOUT. *Oscillator (input and output).* These pins connect a parallel-resonant crystal (24MHz maximum), or an external source to the on-chip clock oscillator and buffer. OSCIN is the input of the oscillator inverter and internal clock generator; OSCOUT is the output of the oscillator inverter.

AV_{DD}. Analog V_{DD} of PLL. This pin must be tied to V_{DD} externally to the ST9294.

V_{DD}. Main Power Supply Voltage (5V±10%)

V_{SS}. Digital Circuit Ground.

P0.0-P0.7, P2.0-P2.5, P3.0-P3.6, P4.0-P4.7, P5.0-P5.1 (J suffix)
P0.0-P0.7, P1.0-P1.7, P2.0-P2.5, P3.0-P3.7, P4.0-P4.7, P5.0-P5.3 (N suffix) *I/O Port Lines (Input/Output, TTL or CMOS compatible).* 31/42 lines grouped into I/O ports, bit programmable under program control as general purpose I/O or as Alternate functions (see next section).

P4.0 - P4.7 are high voltage (12V) open drain

1.2.1 I/O Port Alternate Functions.

Each pin of the I/O ports of the ST9294 may assume software programmable Alternative Functions as shown in the Pin Configuration Drawings. Table 1 shows the Functions allocated to each I/O Port pin.

PIN DESCRIPTION (Continued)

Table 1-1. ST9294 I/O Port Alternative Function Summary

I/O PORT	Name	Function	Alternate Function	Pin Assignment	
Port.bit				9294J	9294N
P0.0		I/O		10	12
P0.1		I/O		9	11
P0.2		I/O		8	10
P0.3		I/O		7	9
P0.4		I/O		6	8
P0.5		I/O		5	6
P0.6		I/O		4	5
P0.7		I/O		3	4
P1.0		I/O		-	52
P1.1		I/O		-	51
P1.2		I/O		-	50
P1.3		I/O		-	49
P1.4		I/O		-	48
P1.5		I/O		-	47
P1.6		I/O		-	46
P1.7		I/O		-	45
P2.0	INT7	I	External Interrupt 7 with Schmitt Trigger	1	2
P2.1	INT5	I	External Interrupt 5 with Schmitt Trigger	42	1
P2.1	AIN1	I	A/D Analog Input 1	42	1
P2.2	INT0	I	External Interrupt 0	41	56
P2.2	AIN2	I	A/D Analog Input 2	41	56
P2.3	INT6	I	External Interrupt 6	40	55
P2.3	SDO	O	MSPI Serial Data Output	40	55
P2.4	NMI	I	Non-Maskable Interrupt	39	54
P2.5	AIN3	I	A/D Analog Input 3	38	53
P3.0	FB	O	Fast Blanking OSD output	18	23
P3.1	R	O	Red Video Colour OSD output	17	22
P3.2	G	O	Green Video Colour OSD output	16	21
P3.3	B	O	Blue Video Colour OSD output	15	20

PIN DESCRIPTION (Continued)

Table 1-1. ST9294 I/O Port Alternative Function Summary

I/O PORT	Name	Function	Alternate Function	Pin Assignment	
Port.bit				9294J	9294N
P3.4		I/O		14	19
P3.5		I/O		13	18
P3.6		I/O		12	17
P3.7		I/O		-	16
P4.0	PWM0	O	PWM Output 0	28	35
P4.1	PWM1	O	PWM Output 1	29	36
P4.2	PWM2	O	PWM Output 2	30	37
P4.3	PWM3	O	PWM Output 3	31	38
P4.4	PWM4	O	PWM Output 4	32	39
P4.5	PWM5	O	PWM Output 5	33	40
P4.6	PWM6	O	PWM Output 6	34	41
P4.7	PWM7	O	PWM Output 7	35	42
P4.7	EXTRG	I	A/D External Trigger	35	42
P5.0	SCK	O	SPI Serial Clock (1)	20	27
P5.0	INT2	I	External Interrupt 2 (1)	20	27
P5.1	SDIO	O	SPI Serial Data Input/Output (1)	19	26
P5.2		I/O		-	25
P5.3		I/O		-	24

Notes.

1. The alternate functions of SCK/INT2 and SDIO may be swapped by using the SWAP Register Function.
2. Schmitt trigger options are available as a mask option for any input pin.

2 CORE ARCHITECTURE

2.1 CORE ARCHITECTURE

The Core or Central Processing Unit (CPU) of the ST9 includes the 8 bit Arithmetic Logic Unit and the 16 bit Program Counter, System and User Stack Pointers. The microcoded Instruction Set is highly optimised for both byte (8 bit) and word (16 bit) data, BCD and Boolean data types, with 14 addressing modes.
Three independent buses are controlled by the Core, a 16 bit Memory bus, an 8 bit Register addressing bus and a 6 bit Interrupt/DMA bus connected to the interrupt and DMA controllers in the on-chip peripherals and the Core. This multiple bus architecture allows a high degree of pipelining and parallel operation, giving the ST9 its efficiency in both numerical calculations and communication with the on-chip peripherals.

2.2 ADDRESS SPACES

The ST9 has three separate address spaces:

- Register File: 240 8-bit registers plus up to 64 pages of 16 bytes each, located in the on-chip peripherals.
- Data memory with up to 64K (65536) bytes
- Program memory with up to 64K (65536) bytes

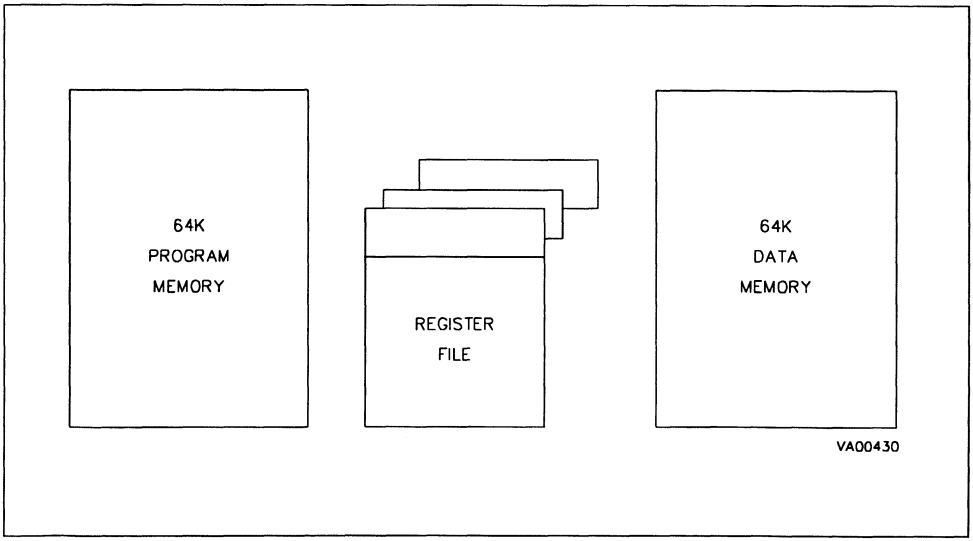
The Data and Program memory spaces will be addressed in further detail in the next section.

2.2.1 Register File

The Register File consists of:

- 224 general purpose registers R0 to R223
- 16 system registers in the System Group (R224 to R239).
- I/O pages depending on the configuration of the ST9, each containing up to 16 registers, with paging facilities based on the top group (R240 to R255).

Figure 2-1. Address Spaces



ADDRESS SPACES (Continued)

Figure 2-2. Register Grouping

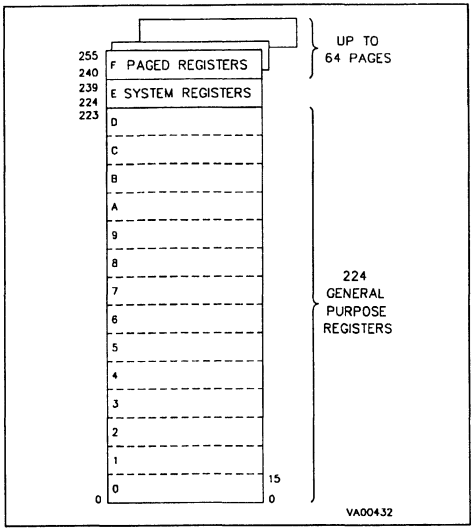


Figure 2-3. Page Pointer Configuration

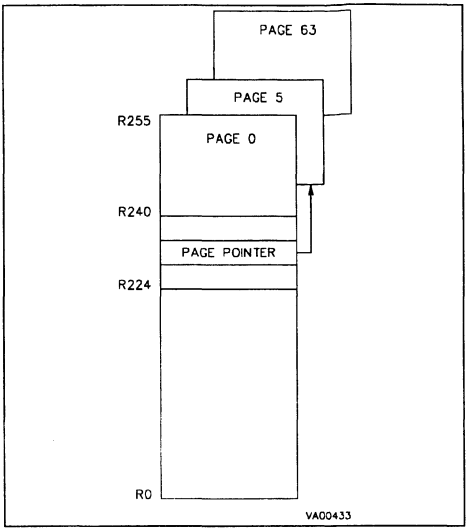
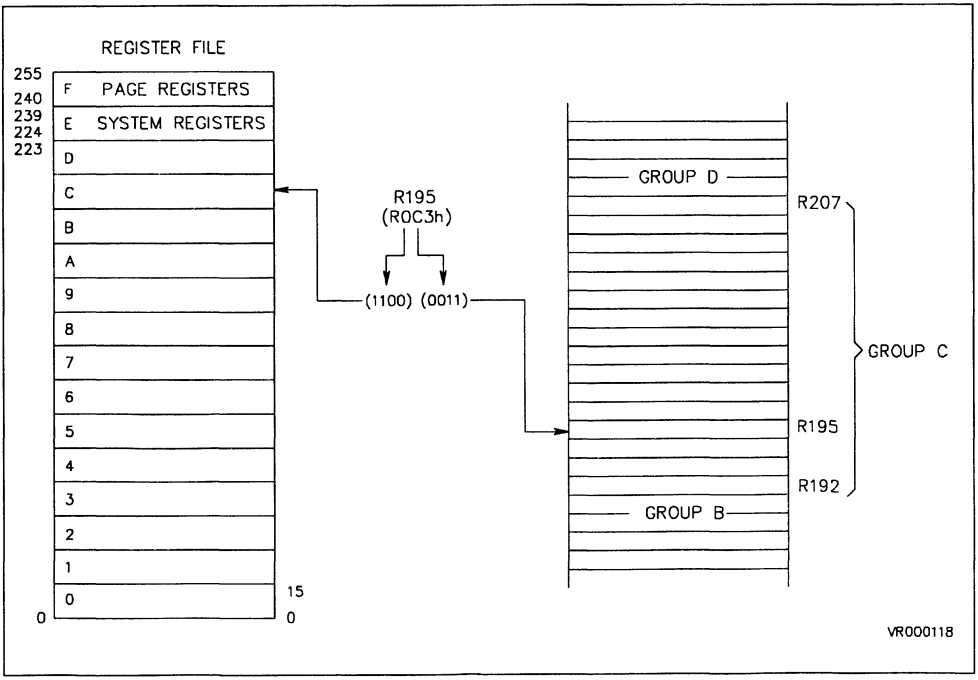


Figure 2-4. Addressing the Register File



ADDRESS SPACES (Continued)

2.2.2 Addressing Registers

All registers in the Register File and pages can be specified by using a decimal, hex or binary address, e.g. R231, RE7h or R11100111b is the same register.
The registers can be referred to by their hexadecimal group address, so that registers R0-R15 form group 0, R160-R175 form group A and so on.

Working Register Addresses

The 8-bit register address is formed by 2 nibbles, for example, for register R195 or RC3h or R11000011, 1100 specifies the 13th group (i.e. group C) and 0011 specifies the 3rd register in that group.

Working registers are addressed by supplying the least significant nibble in the instruction and adding it to the most significant nibble found in the Register Pointer (R233). Working register addressing is shown in Figure 2-4.

System Registers

The 16 system registers at addresses R224 to R239 form Group E.
The system registers are addressable using any of the 4 register addressing modes and the most significant nibble will, in all cases, be 14 (0Eh).

Paged Registers

There are a maximum of 64 pages each containing 16 registers. These are addressed using the register addressing modes with the addition of the Page Pointer register, R234. This register selects the page to be addressed in group F and once set, does not need to be changed if two or more registers on the same page are to be addressed in succession.

Therefore if the Page Pointer, R234, is set to 5, the instructions

```
spp 5
ld R242, r4
```

will load the contents of working register r4 into the third register (R242) of page 5.

These paged registers hold data and control registers related to the on-chip peripherals, and thus the configuration depends upon the peripheral organisation of each ST9 family member. i.e. pages only exist if the peripheral exists.

Available pages are shown in Table 2-2.

2.2.3 Input/Output Ports

The Input/Output ports are located in two areas. The port registers for Ports 0-5 are located at the bottom of the System register group in locations R224 to R229.

Each Port has three associated Control registers, which determine the individual pin modes (I/O, Open-Drain etc). These registers are located in pages 2 and 3.

Table 2-1. Register File Organization

Hex. Address	Decimal Address	Function	Register File Group
F0-FF	240-255	Paged Registers	Group F
E0-EF	224-239	System Registers	Group E
D0-DF	208-223	General Purpose Registers	Group D
C0-CF	192-207		Group C
B0-BF	176-191		Group B
A0-AF	160-175		Group A
90-9F	144-159		Group 9
80-8F	128-143		Group 8
70-7F	112-127		Group 7
60-6F	96-111		Group 6
50-5F	80-95		Group 5
40-4F	64-79		Group 4
30-3F	48-63		Group 3
20-2F	32-47		Group 2
10-1F	16-31		Group 1
00-0F	00-15		Group 0

ADDRESS SPACES (Continued)

Table 2-2. Group F Peripheral Organization

Applicable for ST9294

DEC	HEX	00 00	02 02	03 03	0B 11	28 40	29 41	2A 42	2B 43	3B 59	3E 62	
R255	RFF	SWAP	RESER									RFF
R254	RFE	SPI						RESER		RESER		RFE
R253	RFD		PORT 3									RFD
R252	RFC	WCR							RESER			RFC
R251	RFB		RESER	RESER								RFB
R250	RFA	T/WD			RESER							RFA
R249	RF9		PORT 2			OSD	OSD				RESER	RF9
R248	RF8					CHAR	CHAR	OSD				RF8
R247	RF7		RESER			1 to 16	17 to 32			PWM		RF7
R246	RF6											RF6
R245	RF5	EXT INT	PORT1	PORT 5								RF5
R244	RF4								DATA			RF4
R243	RF3		RESER	RESER				OSD	SLICER			RF3
R242	RF2				SLICE			CHAR				RF2
R241	RF1	RESER	PORT 0	PORT 4	TIMER			33 to 36			A/D	RF1
R240	RF0										CONV	RF0

2.3 SYSTEM REGISTERS

Following is the description of System Registers. For PORT0 to PORT5 Registers, please refer to I/O Port Chapter.

Figure 2-5. System Registers

R239 (EFh)	SYS. STACK POINTER LOW
R238 (EEh)	SYS. STACK POINTER HIGH
R237 (EDh)	USER STACK POINTER LOW
R236 (ECh)	USER STACK POINTER HIGH
R235 (EBh)	MODE REGISTER
R234 (EAh)	PAGE POINTER
R233 (E9h)	REGISTER POINTER 1
R232 (E8h)	REGISTER POINTER 0
R231 (E7h)	FLAGS
R230 (E6h)	CENTRAL INT. CNTL REG
R229 (E5h)	PORT5
R228 (E4h)	PORT4
R227 (E3h)	PORT3
R226 (E2h)	PORT2
R225 (E1h)	PORT1
R224 (E0h)	PORT0

2.3.1 Central Interrupt Control Register

This Register CICR is located in the system Register Group at the address R230 (E6h). Please refer to "INTERRUPT" and "DMA" chapters in order to get the background of the ST9 interrupt philosophy.

CICR R230 (E6h) System Read/Write
Central Interrupt Control Register

Reset Value : 1000 0111

7							0
GCEN	TLIP	TLI	IEN	IAM	CPL2	CPL1	CPL0

b7 = **GCEN:** *Global Counter Enable.* This bit is the Global Counter Enable of the Multifunction Timers. The GCEN bit is ANDed with the CE (Counter Enable) bit of the Timer Control Register (explained in the Timer chapter) in order to enable the Timers when both bits are set. This bit is set after the Reset cycle.

b6 = **TLIP:** *Top Level Interrupt Pending.* This bit is automatically set when a Top Level Interrupt Request is recognized. This bit can also be set by Software in order to simulate a Top Level Interrupt Request.

b5 = **TLI:** *Top Level Interrrupt bit.* When this bit is set, a Top Level interrupt request is acknowledged depending on the IEN bit and the TLNM bit (in Nested Interrupt Control Register). If the TLM bit is reset the top level interrupt acknowledgement depends on the TLNM alone.

b4 = **IEN:** *Enable Interrupt.* This bit, (when set), allows interrupts to be accepted. When reset no interrupts other than the NMI can be acknowledged. It is cleared by interrupt acknowledgement for concurrent mode and set by interrupt return (`iret`). It can be managed by hardware and software (`ei` and `di` instruction).

b3 = **IAM:** *Interrupt Arbitration Mode.* This bit covers the selection of the two arbitration modes, the Concurrent Mode being indicated by the value "0" and the Fully Automatic Nested Mode by the value "1". This bit is under software control.

b2-b0 = **CPL2-CPL0:** *Current Priority Level.* These three bits record the priority level of the interrupt presently under service (i.e. the Current Priority Level, CPL). For these priority levels 000 is the highest priority and 111 is the lowest priority. The CPL bits can be set by hardware or software and give the reference by which following interrupts are either left pending or able to interrupt the current interrupt. When the present interrupt is replaced by one of a greater priority, the current priority value is automatically stored until required.

SYSTEM REGISTERS (Continued)

2.3.2 Flag Register

The Flag Register contains 8 flags indicating the status of the ST9. During an interrupt the flag register is automatically stored in the system stack area and recalled at the end of the interrupt service routine so that the ST9 is returned to the original status. This occurs for all interrupts and, when operating in the nested mode, up to seven versions of the flag register may be stored.

FLAGR R231 (E7h) System Read/Write Flag Register

Reset value: undefined

7							0
C	Z	S	V	DA	H	UF	DP

b7 = **C:** *Carry Flag.* The carry flag C is affected by the following instructions:

Addition (`add, addw, adc, adcw`),
Subtraction (`sub, subw, sbc, sbcw`),
Compare (`cp, cpw`),
Shift Right Arithmetic (`sra, sraw`),
Rotate (`rrc, rrcw, rlc, rlcw, ror, rol`),
Decimal Adjust (`da`),
Multiply and Divide (`mul, div, divws`).

When set, it generally indicates a carry out of the most significant bit position of the register being used as an accumulator (bit 7 for byte and bit 15 for word operations).

The carry flag can be set by the Set Carry Flag (`scf`) instruction, cleared by the Reset Carry Flag (`rcf`) instruction, and complemented (changed to "0" if "1", and vice versa) by the Complement Carry Flag (`ccf`) instruction.

b6 = **Z:** *Zero Flag.* The Zero flag is affected by the following instructions:

Addition (`add, addw, adc, adcw`),
Subtraction (`sub, subw, sbc, sbcw`),
Compare (`cp, cpw`),
Shift Right Arithmetic (`sra, sraw`),
Rotate (`rrc, rrcw, rlc, rlcw, ror, rol`),
Decimal Adjust (`da`),
Multiply and Divide (`mul, div, divws`),
Logical (`and, andw, or, orw, xor, xorw, cpl`),
Increment and Decrement (`inc, incw, dec, decw`),
Test (`tm, tmw, tcm, tcmw, btset`).

In most cases, the Zero flag is set when the register being used as an accumulator register is zero, following one of the above operations.

b5 = **S:** *Sign Flag.* The Sign flag is affected by the same instructions as the Zero flag.

The Sign flag is set when bit 7 (for byte operation) or bit 15 (for word operation) of the register used as an accumulator is one.

b4 = **V:** *Overflow Flag.* The Overflow flag is affected by the same instructions as the Zero and Sign flags.

When set, the Overflow flag indicates that a two's-complement number, in a result register, is in error, since it has exceeded the largest (or is less than the smallest), number that can be represented in twos-complement notation.

b3 = **DA:** *Decimal Adjust Flag.* The Decimal Adjust flag is used for BCD arithmetic. Since the algorithm for correcting BCD operations is different for addition and subtraction, this flag is used to specify which type of instruction was executed last, so that the subsequent Decimal Adjust (`da`) operation can perform its function correctly.
The Decimal Adjust flag cannot normally be used as a test condition by the programmer.

b2 = **H:** *Half Carry Flag.* The Half Carry flag indicates a carry out of (or a borrow into) bit 3, as the result of adding or subtracting two 8-bit bytes, each representing two BCD digits. The Half Carry flag is used by the Decimal Adjust (`da`) instruction to convert the binary result of a previous addition or subtraction into the correct BCD result.
Like the Decimal Adjust flag, this flag is not normally accessed by the user.

b1 = **UF:** *User Flag.* Bit 1 in the flag register (UF) is available to the user, but it must be set or cleared by an instruction.

b0 = **DP:** *Data/Program Memory Flag.* This bit in the flag register indicates which memory area is addressed. Its value is affected by the Set Data Memory (`sdm`) and Set Program Memory (`spm`) instructions.

If the bit is set, the ST9 addresses the Data Memory Area; when the bit is cleared, the ST9 addresses the Program Memory Area. By reading this bit, the user can verify in which memory area the processor is working. The user writes this bit with the `sdm` or `spm` instructions.

SYSTEM REGISTERS (Continued)

2.3.3 Register Pointing Techniques

Two registers, R232 and R233, within the system register group, are available for register pointing. R232 and R233 may be used together as a single pointer for a 16 register working space or separately for two 8 register spaces, in which case R232 becomes Register Pointer 0 (RP0) and R233 becomes Register Pointer 1 (RP1).

The instructions `srp`, `srp0` and `srp1` (the Set Register Pointer instructions) automatically inform the ST9 whether the Register File is to operate with a single 16-register group or two 8-register groups. The `srp0` and `srp1` instructions automatically set the twin 8-register group mode while the `srp` instruction sets the single 16-register group mode. There is no limitation on the order or positions of these chosen register groups other than they must be on 8 or 16 register boundaries.

The addressing of working registers involves use of the Register Pointer value plus an offset value given by the number of the addressed working register.

When addressing a register, the most significant nibble (bits 4-7) gives the group address and the least significant nibble (bits 0-3) gives the register within that group.

REGISTER POINTER 0

RP0 R232 (E8h) System Read/Write Register Pointer 0

Reset Value : undefined

7							0
RG7	RG6	RG5	RG4	RG3	RPS	D1	D0

b7-b3 = **RG7-RG3:** *Register Group number.* These bits contain the number (from 0 to 31) of the group of working registers indicated in the instructions `srp0` or `srp`. When using a 16-register group, a number between 0 and 31 must be used in the `srp` instruction indicating one of the two adjacent 8-register group of working registers used. RG7 is the MSB.

b2 = **RPS:** *Register Pointer Selector.* This bit is set by the instructions `srp0` and `srp1` to indicate that a double register pointing mode is used. Otherwise, the instruction `srp` resets the RPS bit to zero to indicate that a single register pointing mode is used.

b1,b0 = **D1,D0:** These bits are fixed by hardware to zero and are not affected by any writing instruction trying to modify their value.

REGISTER POINTER 1

RP1 R233 (E9h) System Read/Write Register Pointer 1

Reset Value : undefined

7							0
RG7	RG6	RG5	RG4	RG3	RPS	D1	D0

This register is used only with double register pointing mode; otherwise, using single register pointing mode, the RP1R register has to be considered as reserved and not usable as a general purpose register.

b7-b3 = **RG7-RG3:** *Register Group number.* These bits contain the number (from 0 to 31) of the group of 8 working registers indicated in the instructions `srp1`. Bit 7 is the MSB.

b2 = **RPS:** *Register Pointer Selector.* This bit is automatically set by the instructions `srp0` and `srp1` to indicate that a double register pointing mode is used. Otherwise the instruction `srp` reset the RPS bit to zero to indicate that a single register pointing mode is used.

b1,b0 = **D1,D0:** These bits are hardware fixed to zero and are not affected by any writing instruction trying to modify their value.

Note. If working in twin 8-register group mode but only using `srp0` (i.e. only using one 8-register group) the unused register (R233) is to be considered as reserved and not usable as a general purpose register.

The group of registers immediately below the system registers (i.e. group D, R208-R223) can only be accessed via the Register Pointers. To address group D then, it is necessary to set the Register Pointer to group D and then use the addressing procedure for working registers. The programmer is required to remember that the group D should be used as a stacking area. This point is also covered in the Stack Pointers paragraph.

SYSTEM REGISTERS (Continued)

EXAMPLES

Using the Single 16 Register Group

When the system is operating in the single 16-register group mode, the registers are referred to as r0-r15. In this mode, the offset value (i.e. the number of the working register referred to) is supplied in the address (preceded by a small r, e.g. r5) and is added to the Register Pointer 0 value to give the absolute address.

For example, if the Register Pointer contains the value 70h, then working register r7 would have the absolute address, R77h.

In this mode, the single 16-registers group will always start from the lowest even number equal or lower to the number given in the instruction.

Example: `srp #3` is equivalent to `srp #2`.

Using the Twin 8-Register Group

When working in the twin working group mode, the registers pointed by Register Pointer 0 (RP0R), are referred as r0-r7 and those pointed by Register Pointer 1 (RP1R), are referred to as r8-r15, regardless of their absolute addresses. In this mode, when operating with the first 8 working registers (i.e. r0 - r7) the working register number acts as an offset which is added to the value in Register Pointer 0.

So if Register Pointer 0 contains the value 96, then working register 0 has the absolute address 96, working register 5 has the absolute address 101, and so on. The second group of working registers, r8-r15, has the offset values 0 to 7 respectively (i.e. r8 has the offset value 0, r9 has the offset value 1, and so on), this offset value being added to the value in Register Pointer 1.

For example, given that the value in Register Pointer 1 is 32, then working register 12 supplies an offset value of 4 (given by 12 minus 8) to the value in Register Pointer 1 to give an absolute address of 36.

Figure 2-6. Single 16 Register pointing Mode

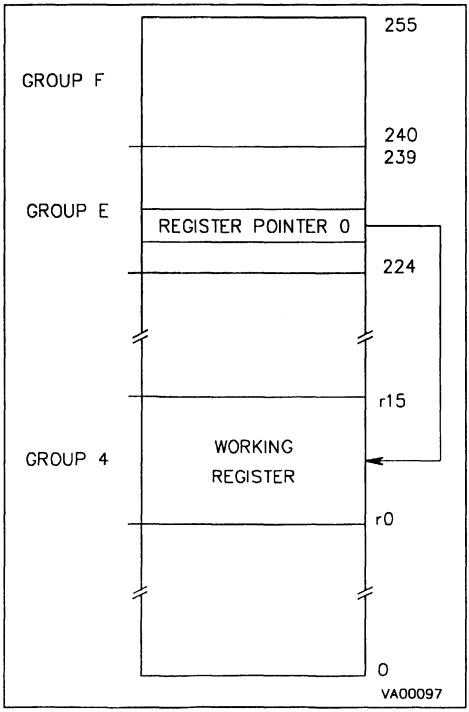
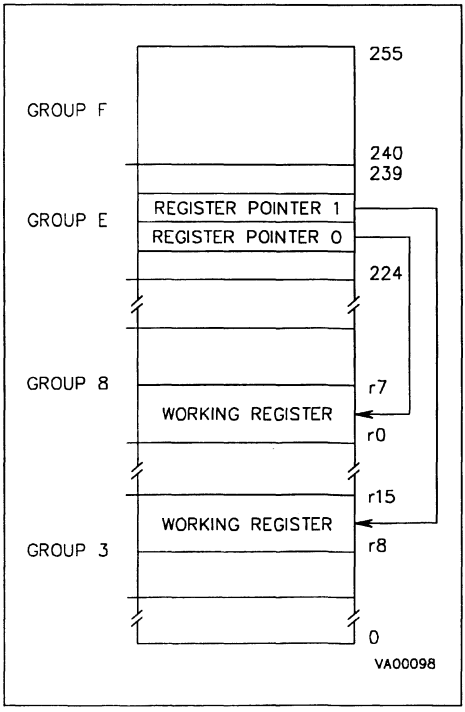


Figure 2-7. Double Register pointing Mode



SYSTEM REGISTERS (Continued)

2.3.4 Page Configuration

The pages are available to be used for the storage of control information (such as interrupt vector pointers) relevant to particular peripherals. There are up to 64 pages (each with 16 registers) based on registers R240-R255. These paged registers are addressable via the page pointer register (PPR), which is system register R234.

To address a paged register the page pointer register (R234) must be loaded with the relevant page number using the `spp` instruction (Set Page Pointer) and subsequently any address from the top (F) group (R240-R255) will be referred to that page.

For example if register 23 contains the value 44, the following sequence loads the third register R242 on page 5 with the value 44.

```
spp 5
ld R242, R23
```

PPR R234 (EAh) System Read/Write Page Pointer Register

Reset value : undefined

7							0
PP7	PP6	PP5	PP4	PP3	PP2	D1	D0

b7-b2 = **PP7-PP2:** *Page Pointer.* These bits contain the number (between 0 to 63) of the page chosen by the instruction `ssp` (Set Page Pointer). PP7 is the MSB of the page address. Once the page pointer has been set, there is no need to refresh it unless a different page is required.

b1-b0 = **D1,D0:** These bits are fixed by hardware to zero and are not affected by any writing instruction trying to modify their value.

PAGE 0 contains the control registers of:

- the external interrupt
- the watchdog timer
- the wait logic states
- the serial peripheral interface (SPI)

2.3.5 Mode Registers

This register MODER is located in the System Register Group at the address 235.

Using this register it is possible:

- to select either internal or external System and User Stack area,
- to manage the clock frequency

MODER R235 (EBh) System Read/Write Mode Register

Reset value : 1110 0000

7							0
SSP	USP	DIV2	PRS2	PRS1	PRS0	BRQEN	HIMP

b7 = **SSP:** *System Stack Pointer.* This bit selects internal (in the Register File) or external (in the external Data Memory) System Stack area, logical "1" for internal, and logical "0" for external. After Reset the value of this bit is "1".

b6 = **USP:** *User Stack Pointer.* Same as bit 7 for the User Stack Pointer;

b5 = **DIV2:** *OSCIN Clock Divided by 2.* This bit controls the divide by 2 circuit which operates on the OSCIN Clock. A logical "1" value means that the OSCIN clock is internally divided by 2, and a logical "0" value means that no division of the OSCIN Clock occurs.

b4-b2 = **PRS2-PRS0:** *ST9 CPUCLK Prescaler.* These bits load the prescaling module of the internal clock (INTCLK). The prescaling value selects the frequency of the ST9 clock, which can be divided by 1 to 8. See Clock chapter for more information.

b1 = **BRQEN:** *Bus Request Enable.* This bit must be held to "0".

b0 = **HIMP:** *High Impedance Enable.* This bit must be held to "0".

SYSTEM REGISTERS (Continued)

2.3.6 Stack Pointers

There are two separate, double register stack pointers available (named System Stack Pointer and User Stack Pointer), both of which can address registers or memory.

The stack pointers point to the bottom of the stacks which are filled using the `push` commands and emptied using the `pop` commands. The stack pointer is automatically pre-decremented when data is "pushed in" and post-incremented when data is "popped out".

For example, the register address space is selected for a stack and the corresponding stack pointer register contains 220. When a byte of data is "pushed" into the stack, the stack pointer register is decremented to 219, then the data byte is "loaded" into register 219. Conversely, if a stack pointer register contains 189 and a byte of data is "popped" out, the byte of data is then extracted from the stack and then the stack pointer register is incremented to 190.

The `push` and `pop` commands used to manage the system stack area are made applicable to the user stack by adding the suffix U, while to use a stack instruction for a word a W is added.

For example `push` inserts data into the system stack, but an added U indicates the user stack and W means a word, so the instruction `pushuw` loads a word into the bottom of the user stack.

If the User Stack Pointer register contains 223 (working in register space) the instruction `pushuw` will decrement User Stack Pointer register to 222 and then load a word into register R222 and R221.

When bytes (or words) are "popped out" the values in those registers are left unchanged until fresh data is loaded into those locations. Thus when data is "popped" out from a stack area, the stack content remains unchanged.

Note. Stacks must not be located in the pages or the system register area.

The System Stack area and The System Stack Pointer

The System Stack area is used for the storage of temporarily suspended system and/or control registers, i.e. the Flag register and the Program counter, while interrupts are being serviced. For subroutine execution only the Program Counter needs to be saved in the System stack area.

There are two situations when this occurs automatically, one being when an interrupt occurs and the other when the instruction call subroutine is used. When the system stack area is in the Register File, the stack pointer, which points to the bottom of the stack, only needs one byte for addressing, in which case the System Stack Pointer Low Register (R239) is sufficient for addressing purposes. As a result the System Stack Pointer High Register (R238) becomes redundant BUT must be considered as reserved (please refer also to "spurious" memory access section). Clearly when the stack is external a full word address is necessary and so both registers are used to point, the even register providing the MSB and the odd register providing the LSB.

The User Stack area and User Stack Pointer

The User Stack area is completely free from all interference from automatic operations and so it provides a totally user controlled stacking area, that area being in any part of the memory which is of a RAM nature, or the first 14 groups of the general Register File i.e. not in the System register or Paged group.

The User Stack Pointer consists of two registers, R236 and R237, which are both used for addressing an external stack, while, when stacking in the Register File, the User Stack Pointer High Register, R236, becomes redundant but must be considered as reserved.

SYSTEM REGISTERS (Continued)

Stack location

Care is necessary when managing stacks as there is no limit to stack sizes apart from the bottom of any address space in which the stack is placed. Consequently programmers are advised to use a stack pointer value as high as possible, particularly when using the Register File as a stacking area. This will also benefit programmers who may locate the stacks in group D using, for example the instruction `ld R237, #223` which loads the value 223 into the User Stack Pointer Low Register. The Programmer will not need to remember to set the Register Pointer to 208 to gain access to registers in the D-group, a problem outlined in Register Pointing Techniques paragraph.

Stacks may be located anywhere in the first 14 groups of the Register File (internal stacks) or the data memory (external stacks). It is not necessary to set the data memory using the instruction `sdm` as external stack instructions automatically use the data memory.

Figure 2-8. System and/or User Stack in Register Stack Mode

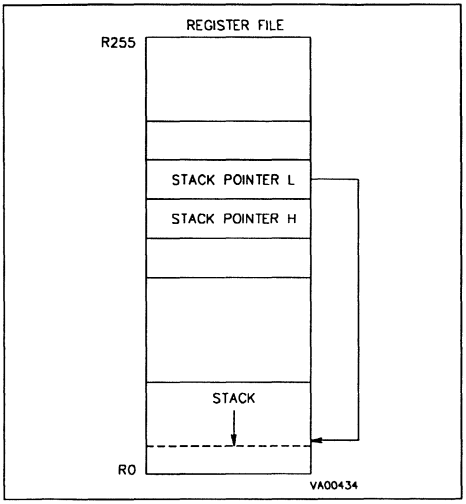
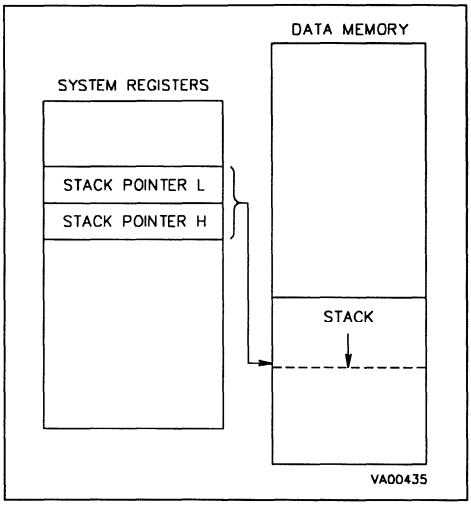


Figure 2-9. System and/or User Stack in Memory Stack Mode



USP R236 (ECh) System Read/Write
User Stack Pointer High Byte

Reset value: undefined

7							0
USP15	USP14	USP13	USP12	USP11	USP10	USP9	USP8

USP R237 (EDh) System Read/Write
User Stack Pointer Low Byte

Reset value: undefined

7							0
USP7	USP6	USP5	USP4	USP3	USP2	USP1	USP0

SSP R238 (EEh) System Read/Write
System Stack Pointer High Byte

Reset value: undefined

7							0
SSP15	SSP14	SSP13	SSP12	SSP11	SSP10	SSP9	SSP8

SSP R239 (EFh) System Read/Write
System Stack Pointer Low Byte

Reset value: undefined

7							0
SSP7	SSP6	SSP5	SSP4	SSP3	SSP2	SSP1	SSP0

NOTES

3 MEMORY

3.1 INTRODUCTION

The memory of the ST9294 is functionally divided into two areas, the Register File and Memory. The Memory may optionally be divided into two spaces, Program Memory for Program code and Data Memory for Data.

The memory spaces are selected by the execution of the SDM and SPM instructions (Set Data Memory and Set Program Memory, respectively). There is no need to use either of these instructions again until the memory area required is to be changed.

3.1.1 Program Space

The Program memory space of the ST9294 consists of 24K bytes of on-chip ROM (addressed from 0 to 5FFFh) and 640 bytes of on-chip RAM (addressed from 6000h to 627Fh). The first 256 memory locations from address 0 to 0FFh (hexadecimal) hold the Reset Vector, the Top-Level (Pseudo Non-Maskable) interrupt, the Divide by Zero Trap vector and, optionally, the interrupt vector table for use with the on-chip peripherals and the external interrupt sources. Each vector is contained in two consecutive byte locations, the high order address held in the lower (even) byte, the low order address held in the upper (odd) byte, forming the address which is loaded into the Program Counter when selected by the interrupt vector provided by the interrupt source. This should point to the relevant Interrupt Service routine provided by the User for immediate response to the interrupt.

3.1.2 Data Space

The ST9294 addresses the 640 bytes of on-chip RAM memory from addresses 6000h to 627Fh in both Program and Data Space. On-chip general purpose Registers may be used as additional RAM memory for minimum chip count systems.

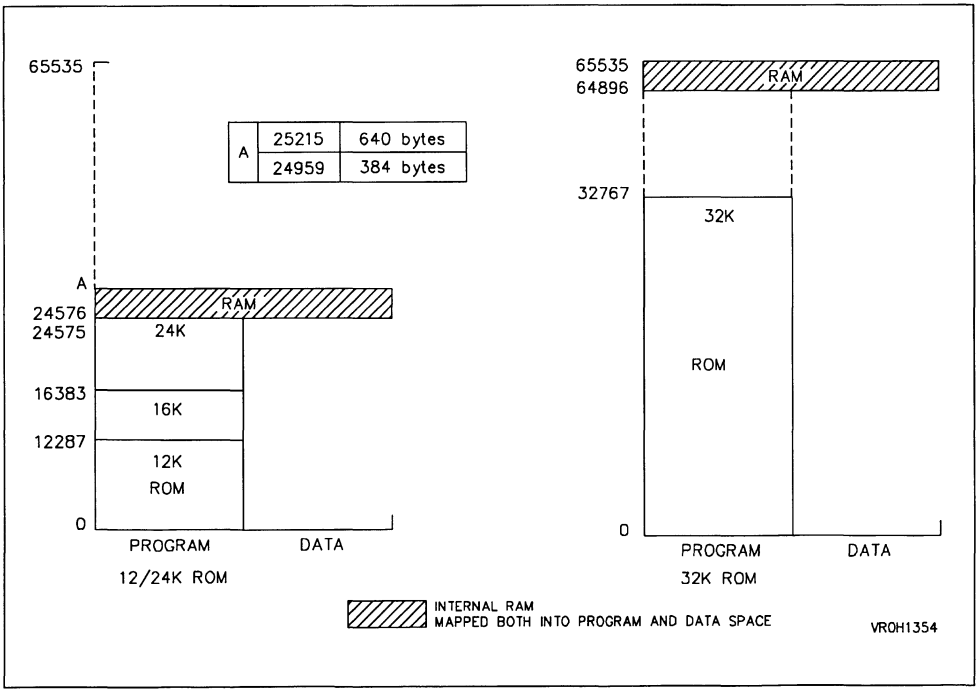
The Data Space is selected by the execution of the SDM instruction. All subsequent memory references will access the Data Space. When a separate Data Space is not required, data may stored in RAM or ROM memory within the Program Space.

MEMORY (Continued)

Table 3-1. ROM and RAM Address Configuration

Device Suffix	ROM Size (Bytes)	ROM Addresses		RAM Size (Bytes)	RAM Addresses	
J6/N6	32K	0 - 32767	dec	640	64896 - 65535	dec
		0000 - 7FFF	hex		FD80 - FFFF	hex
J5/N5	24K	0 - 24575	dec	640	24576 - 25215	dec
		0000 - 5FFF	hex		6000 - 627F	hex
J4/N4	24K	0 - 24575	dec	384	24576 - 24959	dec
		0000 - 5FFF	hex		6000 - 617F	hex
J3/N3	16K	0 - 16383	dec	640	24576 - 25215	dec
		0000 - 3FFF	hex		6000 - 627F	hex
J2/N2	16K	0 - 16383	dec	384	24576 - 24959	dec
		0000 - 3FFF	hex		6000 - 617F	hex
J1/N1	12K	0 - 12287	dec	640	24576 - 25215	dec
		0000 - 2FFF	hex		6000 - 627F	hex

Figure 3-1. ST9294 Memory Map



4 INTERRUPTS

4.1 INTRODUCTION

The ST9 responds to peripheral events and external events through its Interrupt channels. When such an event occurs, if previously enabled and according to a priority mechanism, the current program execution can be suspended to allow the ST9 to execute a specific response routine. If the event generates an interrupt request, the current program status is saved after the current instruction is completed and the CPU control passes to the Interrupt Service Routine.

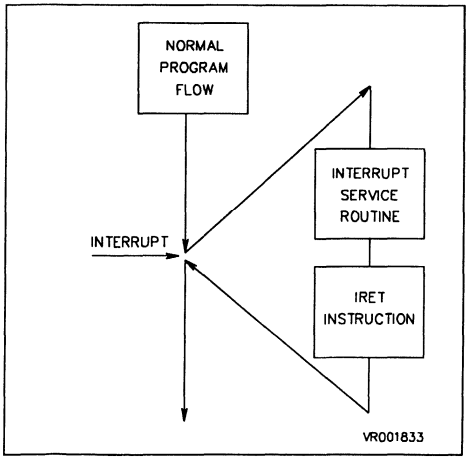
The ST9 CPU can receive requests from the following type of sources:

- On-chip peripherals
- External pins
- Top-Level Pseudo-non-maskable interrupt

According to the on-chip peripheral features, an event occurrence can generate an Interrupt request depending on the selected mode.

Up to eight external interrupt channels, with programmable input trigger edge, are available. In addition, a dedicated interrupt channel, set to the Top-level priority, can be devoted either to the external pin NMI (to provide a Non-Maskable-Interrupt) or to the Timer/Watchdog. Interrupt service routines are addressed through a vector table mapped in Program Memory.

Figure 4-1. Interrupt Flow



4.2 INTERRUPT VECTORIZATION

The ST9 implements an interrupt vectoring structure that allows the on-chip peripheral to identify the location of the first instruction of the Interrupt Service Routine (IVR) automatically.

When the interrupt request is acknowledged, the peripheral interrupt module provides, through its Interrupt Vector Register (IVR), a vector to point into the vector table of locations containing the start addresses of the Interrupt Service Routines (defined by the programmer).

Each peripheral has a specific IVR mapped within its Register File pages.

The Interrupt Vector table, containing the list of the addresses of the Interrupt Service Routines, is located in the first 256 locations of the Program Memory. The first 6 locations of the Program Memory are reserved for:

Address Content

0	Address high of Power on Reset routine
1	Address low of Power on Reset routine
2	Address high of Divide by zero trap Subroutine
3	Address low of Divide by zero trap Subroutine
4	Address high of Top Level Interrupt routine
5	Address low of Top Level Interrupt routine

With one Interrupt Vector register, it is possible to address more interrupt service routines; in fact, several peripherals share the same interrupt vector register among several interrupt channels. The most significant bits of the vector are user programmable to define the base vector address inside the vector table in the program memory, the least significant bits are controlled by the interrupt module in hardware to select the specific vector.

Note: The first 256 locations of the program memory can contain program code. Other than the Reset vector, they are not exclusively reserved to the vector table.

Warning. *Although the Divide by Zero Trap operates as an interrupt, the FLAG Register is not pushed onto the system Stack automatically. As a result it must be regarded as a subroutine, and the acknowledge routine must end with the* RET *instruction.*

INTERRUPT VECTORIZATION (Continued)

Figure 4-2. Vectors and Associated Routines

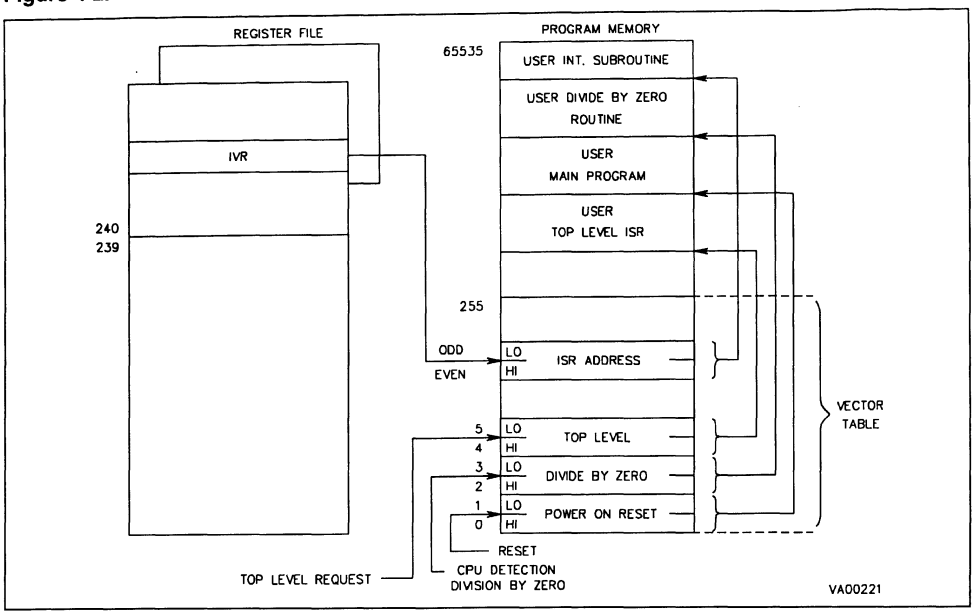
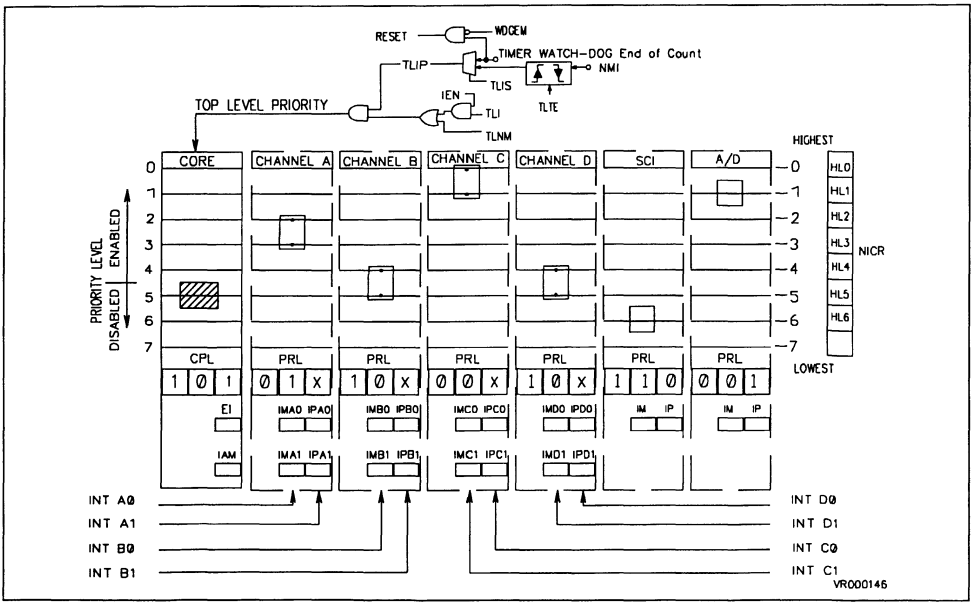


Figure 4-3. Interrupt Architecture, Example of priority Allocations



4.3 INTERRUPT PRIORITY LEVEL ARCHITECTURE

The ST9 supports a fully programmable interrupt priority structure. Figure 4-4 shows a conceptual description.

9 priority levels are available to define the channel priority relationship. Each channel has a 3 bit field, PRL (Priority Level), that defines its priority level among 8 programmable levels. The ninth level (Top Level Priority) is reserved for the Timer/Watchdog or the External Pseudo Non-Maskable Interrupt. The On-chip peripheral channel and the eight external interrupt sources can be programmed within eight priority levels: level 7 has the lowest priority, level 0 has the highest priority.

If several units are located at the same priority level, an internal daisy chain, fixed for each ST9 device, defines the priority relationship within that level.

The PRL bits are used to define the priority level for interrupt requests.

Top level priority interrupt (highest) can be assigned either to the external Pseudo Non-Maskable interrupt or to the internal Timer/Watch-Dog. An Interrupt service routine at this level cannot be interrupted in any arbitration mode. Its mask can be both maskable (TLI) or non-maskable (TLNM).

4.4 PRIORITY LEVEL ARBITRATION

The 3 bits of CPL (Current Priority Level) in the Central Interrupt Control Register contain the priority of the currently running program (CPU priority). CPL is set to 7 (lowest priority) upon reset and can be modified during program execution either by software or automatically by hardware according to the selected Arbitration Mode.

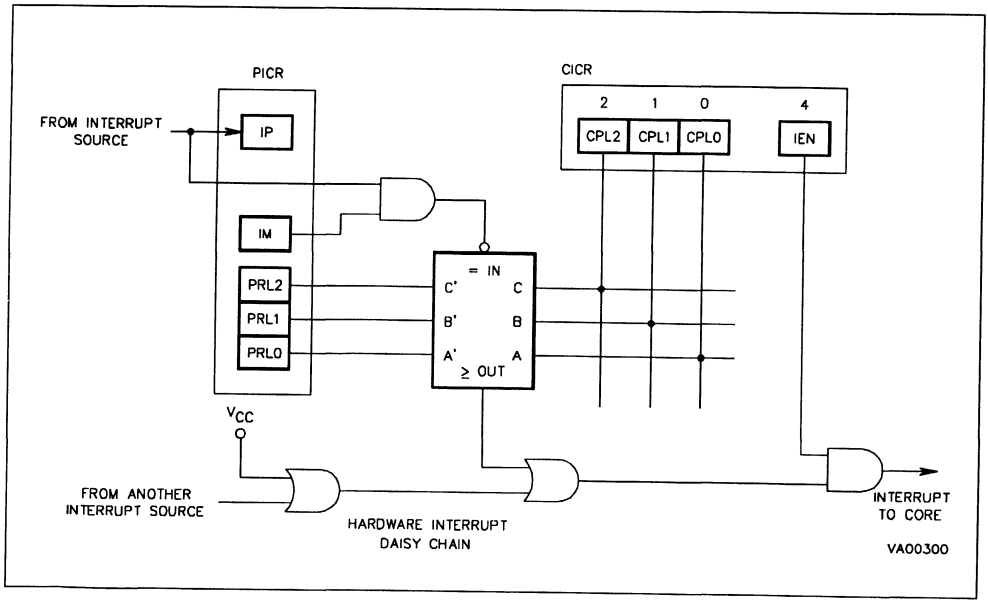
During every instruction an arbitration phase is made between every channel capable of generating an Interrupt, each priority level is compared to all the other requests. If the highest priority request is an interrupt, it must be *higher* than the CPL value in order to be acknowledged.

The priority of the Top Level Interrupt overrides every other priority.

If two or more requests occur at the same instant of time and at the same priority level, an on-chip daisy chain, specific to every ST9 version, selects the channel with the highest position in the chain. The position in the chain is shown in table 4-1.

ST9 provides two interrupt arbitration modes: Concurrent and Nested modes. The Concurrent mode is the standard interrupt arbitration mode while the Nested mode improves the effective interrupt response time when a nesting of the service routines is required according to the request priority levels.

Figure 4-4. Interrupt Logic



PRIORITY LEVEL ARBITRATION (Continued)

The control bit IAM (CICR.3) selects the Concurrent Arbitration mode (when reset to "0") or the Nested Arbitration Mode (when set to "1").

Table 4-1. Daisy Chain Priorities

Applicable for ST9294

Highest Position	INTA0
	INTA1
	INTB0
	INTB1
	INTC0
	INTC1
	INTD0
Lowest Position	INTD1

Warning: *Although the divide by Zero Trap operates as an interrupt, the FLAG Register is not pushed onto the system Stack automatically. As a result it must be regarded as a subroutine, and the acknowledge routine must end with the* `RET` *instruction.*

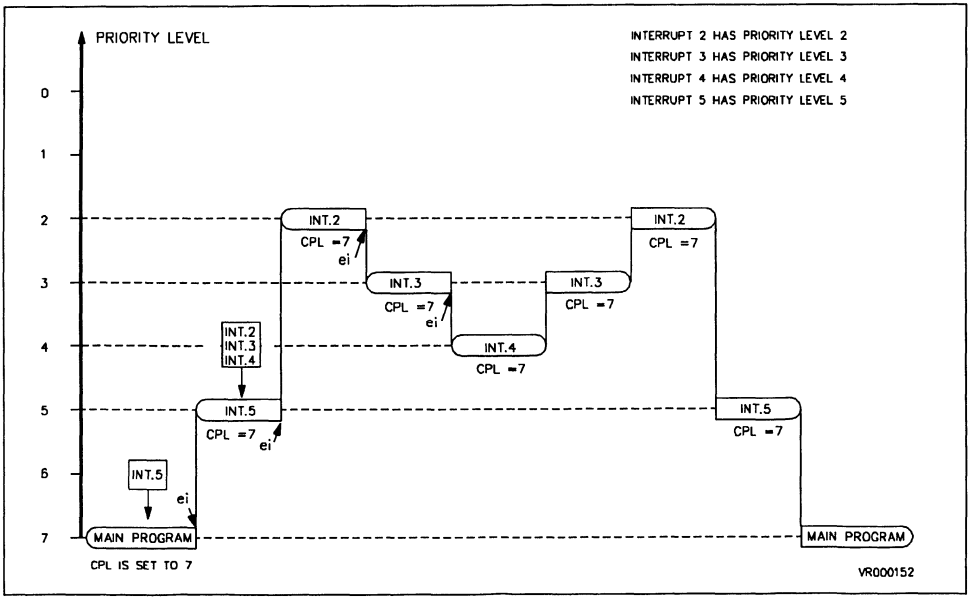
4.4.1 Concurrent Mode

This mode is selected when the IAM bit is cleared (reset condition). The arbitration phase, performed during every instruction, selects the request with the highest priority level.

If the highest priority request is an interrupt request and its priority value is higher than the Current Priority Value CICR.2,1,0 (R230.2,1,0), the interrupt request will be acknowledged at the end of the current instruction. The interrupt Machine Cycle performs the following steps:

- 1. Disables all the maskable interrupt requests by clearing CICR.IEN
- 2. Pushes the PC low byte into the system stack
- 3. Pushes the PC high byte into the system stack
- 4. Pushes the Flag register into the system stack
- 5. Loads the PC with the 16-bit vector stored in the Vector Table, pointed to by the Interrupt Vector Register (IVR).

Figure 4-5. Example of a Sequence of Interrupt Requests with :
- Concurrent mode
- EI set to 1 during the interrupt routine execution



PRIORITY LEVEL ARBITRATION (Continued)

The Interrupt Service Routine must be concluded with the `iret` instruction. The `iret` instruction executes the following operations:

- 1. Pops off the Flag register from the system Stack
- 2. Pops off PC high byte from the system Stack
- 3. Pops off PC low byte from the system Stack
- 4. Enables all the un-masked Interrupts, by setting the CICR.IEN bit

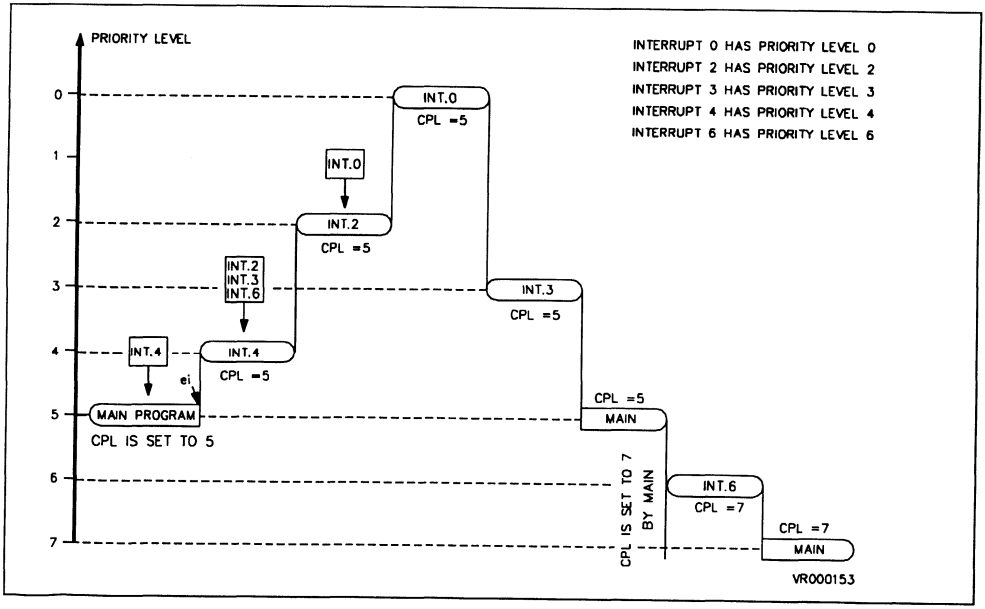
The suspended program execution is thus recovered at the interrupted instruction. All pending interrupts existing, or having occurred during the interrupt service routine execution, remain pending until the Enable Interrupt instruction (even if it is executed during the interrupt service routine).

NOTE: When Concurrent mode is selected, the source priority level is meaningful only during the arbitration phase, where it is compared to all the other priority levels and the CPL, but no trace is kept of its value during the Interrupt Service Routine. Therefore, if other requests are issued, once the global CICR.IEN is enabled again, they will be acknowledged regardless of the Interrupt Service Routine priority value; if no care is taken by the programmer, unpleasant side effects can take place.

A typical case is the following: 3 pending requests with different priority levels (ie 2,3,4) generate requests at the same time (because the associated events occurred during the same instruction). The three interrupt service routines set Interrupt Enable (IEN, CICR.4) by the `ei` instruction at the beginning of the routine to avoid a high interrupt response time to requests with a priority higher than the one under service (usually, the higher the priority, the sooner the routine must be executed). Unfortunately, what will happen in this case is that the three interrupt servicing routines will be executed exactly in the opposite order of their priority. Interrupt routine level 2 will be acknowledged first, then, when the `ei` instruction is executed, it will be interrupted by interrupt routine level 3, which itself will be interrupted by interrupt routine level 4. When interrupt routine level 4 is completed, interrupt routine level 3 will be recovered and finally, interrupt routine level 2.

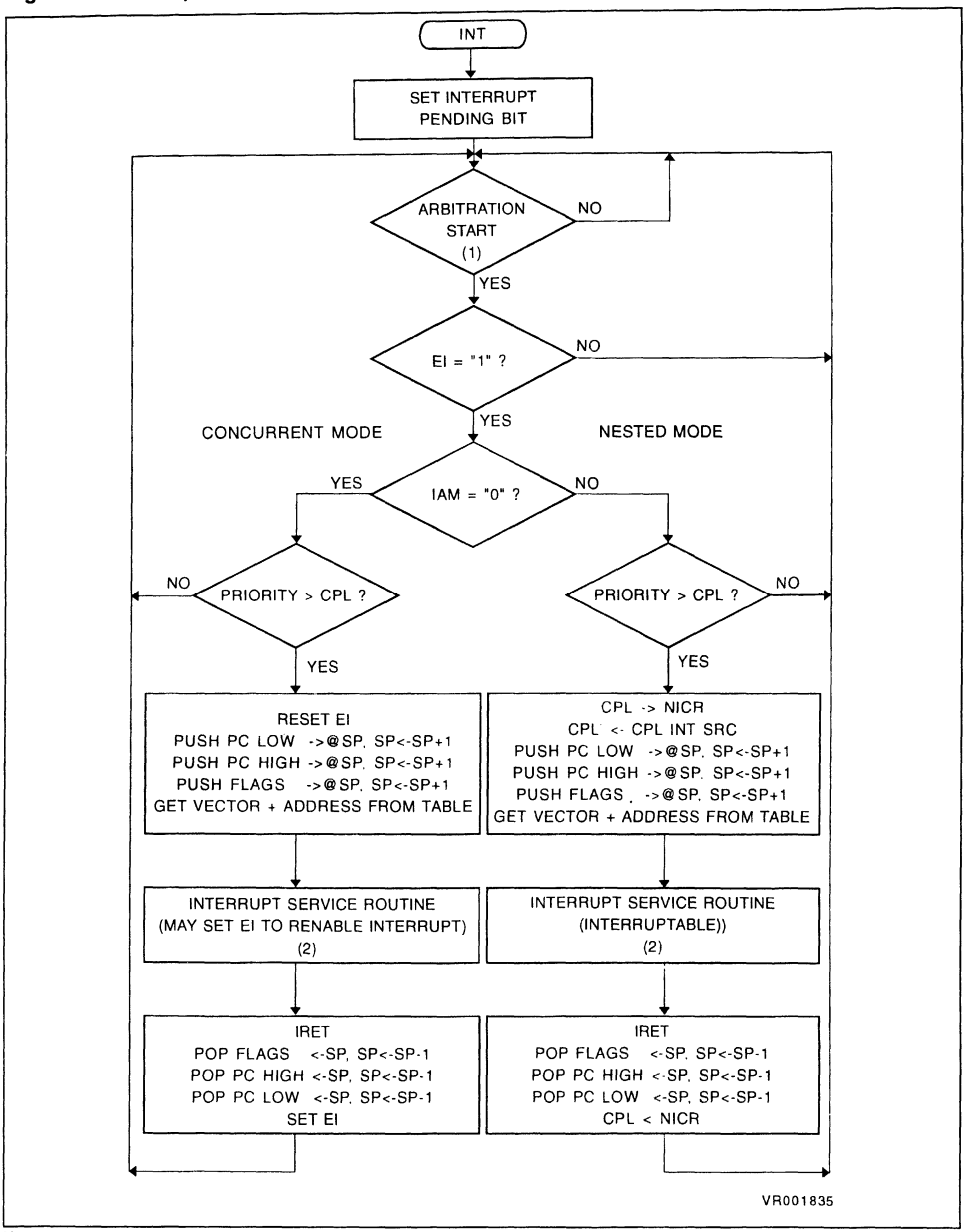
Therefore, **it is recommended, in concurrent mode, to avoid the insertion of the `ei` instruction in the interrupt subroutine**, which can trigger this LIFO (Last In, First Out) sequence of interrupt processing.

Figure 4-6. Example of a Sequence of Interrupt Requests with :
- Concurrent mode
- EI unchanged by the interrupt routines



PRIORITY LEVEL ARBITRATION (Continued)

Figure 4-7. Interrupt Mode Flow-Chart



Notes:
1. The interrupt arbitration starts 6 CPUCLK cycles before the end of execution of each instruction (5 cycles during WFI).
2. Clear interrupt pending bit

PRIORITY LEVEL ARBITRATION (Continued)

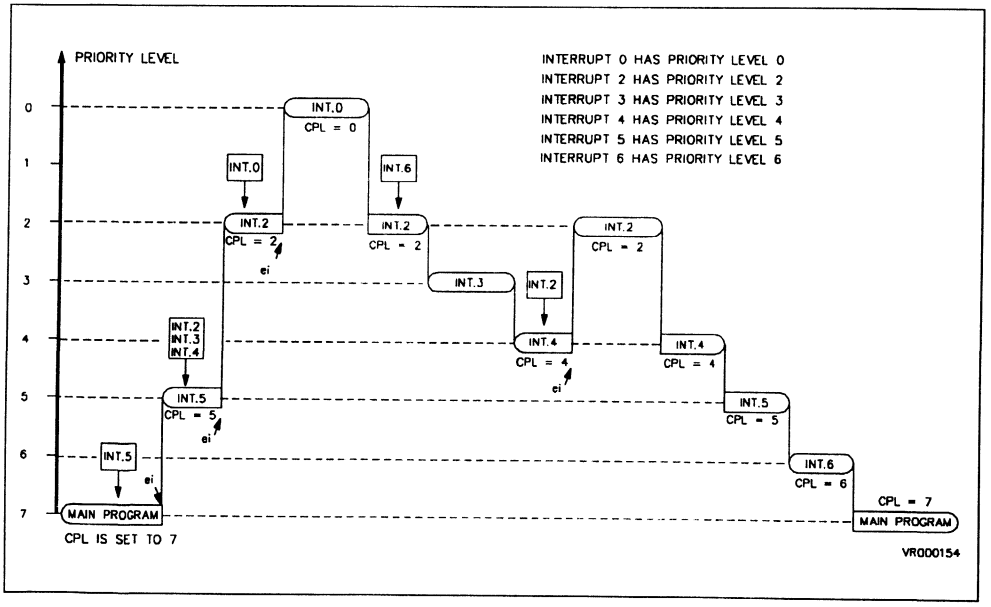
4.4.2 Nested Mode

The difference of the Nested mode to the Concurrent mode consists of the modification of the CPL value during the interrupt processing. The arbitration phase is basically identical to the concurrent Mode, however once the request is acknowledged, the current CPL value is saved in the Nested Interrupt Control Register (NICR, R247 page 0) by setting the NICR bit corresponding to the CPL value (i.e. if the CPL is 3, NICR.3 bit will be set). The CPL value is then updated with the Priority value of the request just acknowledged, in this way the next arbitration cycle will be performed against the priority level of the Service Routine in progress.

The Interrupt Machine Cycle will perform the following steps:

- Disable all the maskable interrupts by clearing IEN
- Save the CPL value into the special stack NICR to hold the priority level of the suspended routine
- Store in CPL the priority level of the acknowledged routine, so that the next request priority will be compared with the one of the routine under service
- Push the PC-low byte into the System Stack
- Push the PC-high byte into the System Stack
- Push the Flag Register into the System Stack
- Load the PC with the vector pointed by IVR.

Figure 4-8. Example of a Sequence of Interrupt Requests with :
- Nested mode
- EI set to 1 during the interrupt routine execution



PRIORITY LEVEL ARBITRATION (Continued)

The `iret` Interrupt Return instruction executes the following steps:

- 1. Pop off the Flag Register from the System Stack
- 2. Pop off the PC-high byte from the System Stack
- 3. Pop off the PC-low byte from the System Stack
- 4. Enable all the unmasked interrupts by setting the IEN bit
- 5. Recover the interrupted routine priority level by popping the value from the special register (NICR) and by copying it into CPL.

The suspended execution is thus recovered at the interrupted instruction.

REMARKS

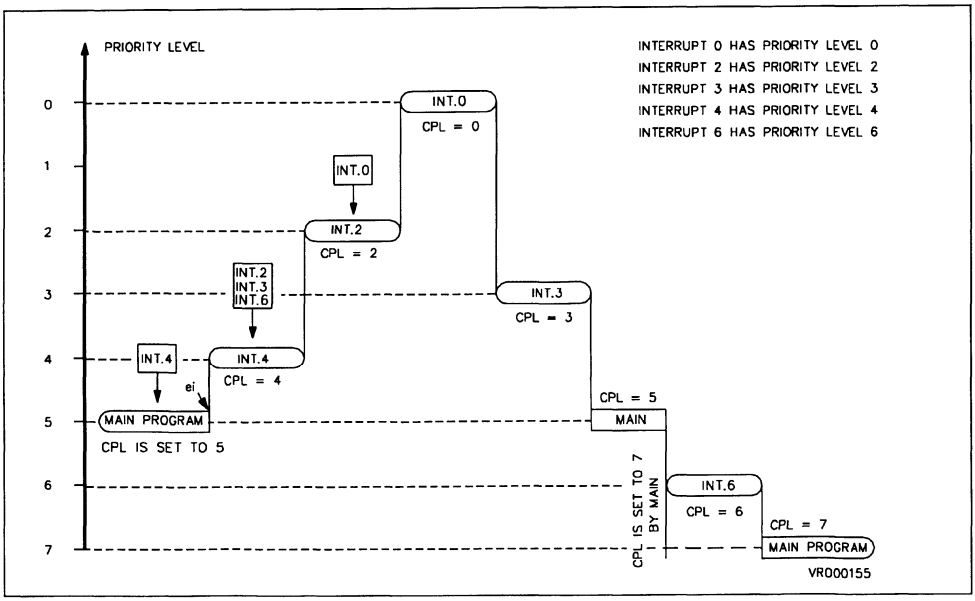
1) Dynamic priority level modification: the main program and routines can be specifically prioritized. Since CPL is represented by 3 bits in a read/write register, it is possible to modify dynamically the current priority value during the program execution. This means that a critical section can have a higher priority with respect to other interrupt requests. Furthermore it is possible to prioritize even the Main Program execution by modifying CPL during its execution.

2) Maximum number of nestings: No more than 8 routines can be nested. If an interrupt routine at level N is being serviced, no other Interrupts located at level N can interrupt it. This guarantees a maximum number of 8 nested levels including the Top Level Interrupt request.

3) Priority level 7: Interrupt requests at level 7 cannot be acknowledged as their priority cannot be higher than the CPL value. This can be of use in a fully polled interrupt environment.

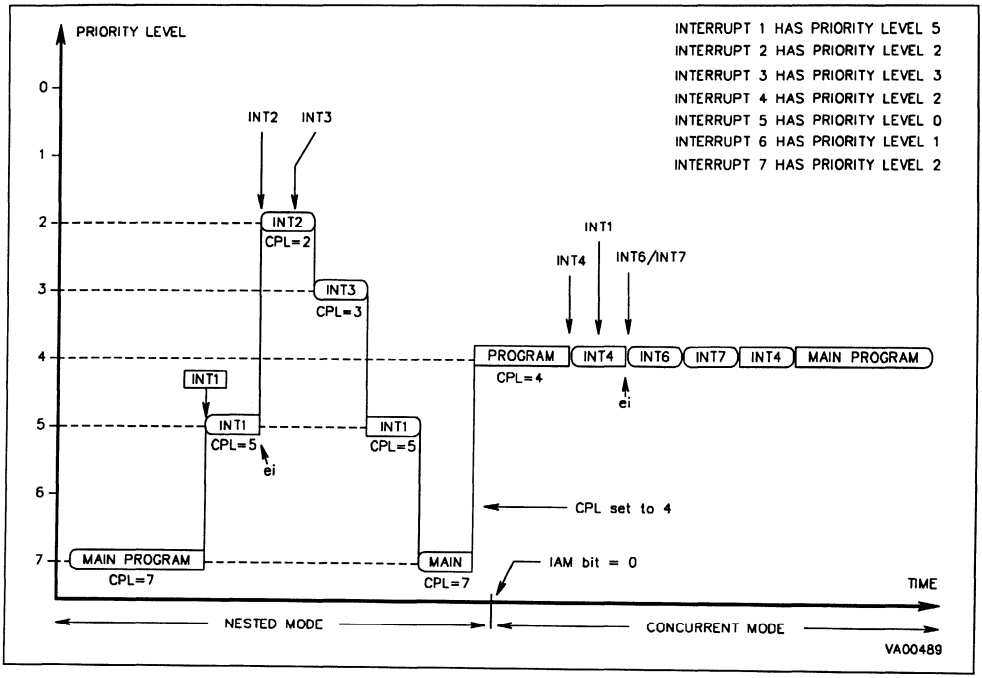
A nested/concurrent mode sequence is given on Figure 4-10. This example clearly shows that Nested and Concurrent modes are defined by the user. Note that here the Y axis is referenced by CPL, instead of the source priority level, and *that Interrupt 1 stays pending*, having a priority level lower than CPL.

Figure 4-9. Example of a Sequence of Interrupt Requests with :
- Nested mode
- EI unchanged by the interrupt routiness



PRIORITY LEVEL ARBITRATION (Continued)

Figure 4-10. Example of a Nested and Concurrent Mode Sequence



4.5 EXTERNAL INTERRUPTS

The standard ST9 core contains 8 external interrupts sources grouped into four pairs.

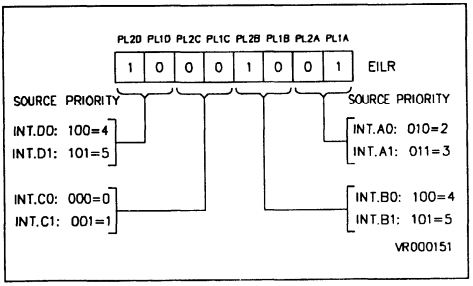
Table 4-2. External Interrupt Channel Grouping

External Interrupt	Channel
INT7 INT6	INTD1 INTD0
INT5 INT4	INTC1 INTC0
INT3 INT2	INTB1 INTB0
INT1 INT0	INTA1 INTA0

Each source has a trigger control bit TEA0,..TED1 (R242,EITR.0,..,7 Page 0) to select triggering on the rising or falling edge of the external pin. If the Trigger control bit is set to "1", the corresponding pending bit IPA0,..,IPD1 (R243,EIPR.0,..,7 Page 0) is set on the input pin rising edge, if it is cleared, the pending bit is set on the falling edge of the input pin. Each source can be individually masked through the corresponding control bit IMA0,..,IMD1 (EIMR.7,..,0). See Figure 4-12.

The priority level of the external interrupt sources can be programmed among the eight priority levels with the control register EIPLR (R245). The priority level of each pair is software defined using the bits PRL2,PRL1. For each pair, the even channel (A0,B0,C0,D0) of the group has the even priority level and the odd channel (A1,B1,C1,D1) has the odd (lower) priority level. Figure 4-11 shows an example of priority levels.

Figure 4-11. Priority Level Examples



In the ST9294:

- The source of the interrupt channel A0 can be selected between the external pin INT0 (when IA0S = "1", the reset value) or the On-chip Timer/Watchdog peripheral (when IA0S = "0").
- The source of the interrupt channel A1 is the End of Count condition of the on-chip Slice Timer.
- The source of the interrupt channel B0 can be selected between the external pin INT2 (when (SPEN,BMS)=(0,0)) or the on-chip SPI peripheral.
- The source of the interrupt channel B1 is the End of Conversion condition of the on-chip A/D Converter.
- The source of the interrupt channel C0 is the Display Swap condition of the on-chip On Screen Display.

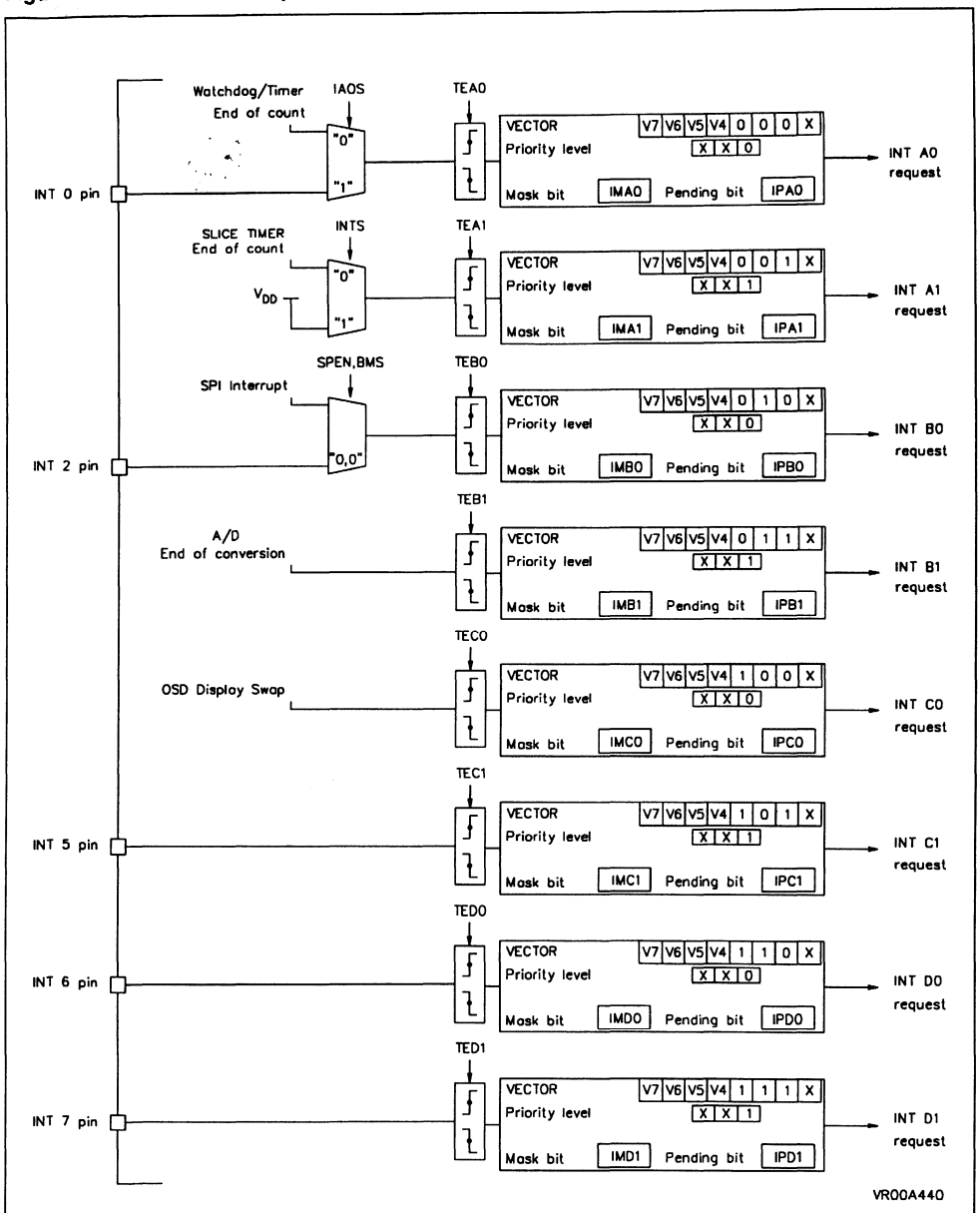
All other interrupt channels have an input pin as source, however, the input line may be multiplexed with an on-chip peripheral I/O or connected to an input pin that performs also other function (as in the case of the handshake feature).

Table 4-3. External Interrupt Source Selection

INT0	INTA0
Timer/Watchdog End of Count	
Slice Timer End of Count	INTA1
INT2	INTB0
SPI Interrupt	
A/D End of Conversion	INTB1
OSD Display Swap	INTC0

EXTERNAL INTERRUPTS (Continued)

Figure 4-12. External Interrupts Control Bits and Vectors



4.6 TOP LEVEL INTERRUPT

The Top Level Interrupt channel can be assigned either to the external pin NMI or to the Timer/Watchdog according to the status of the control bit EIVR.TLIS (R246.2, Page 0). If this bit is high (the reset condition) the source is the external pin NMI, if it is low, the source is the Timer/ Watchdog End Of Count. When the source is the NMI external pin, the control bit EIVR.TLTEV (R246.3; Page 0) selects between the rising (if set) or falling (if cleared) edge generating the interrupt request. When the selected event occurs, the CICR.TLIP bit (R230.6) is set. Depending on the mask situation, a Top Level Interrupt request may be generated. Two kinds of masks are available, a Maskable mask and a Non-Maskable mask. The first mask is the bit CICR.TLI (R230.5): it can be set or cleared to enable or disable respectively the Top Level Interrupt request. If it is enabled, the global Enable Interrupt bit CICR.IEN (R230.4) must also be enabled in order to allow a Top Level Request.

The second mask NICR.TLNM (R247.7) is a set-only mask. Once set, it enables the Top Level Interrupt request independently of the value of CICR.IEN and it cannot be cleared by program. Only the processor RESET cycle can clear this bit.

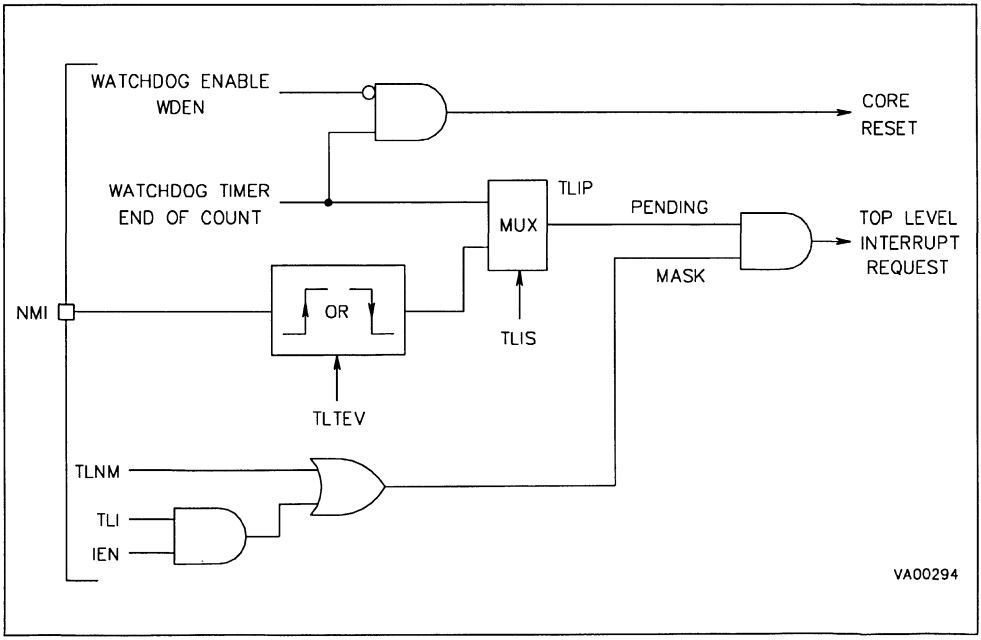
The Top Level Interrupt Service Routine cannot be interrupted by any other interrupt request, in any arbitration mode, even by another Top Level Interrupt request.

Warning. *The interrupt machine cycle of the Top Level Interrupt does not clear the CICR.IEN bit, and the corresponding* `iret` *does not set it.*

4.7 ON-CHIP PERIPHERAL INTERRUPTS

The general structure of the peripheral interrupt unit is described here, however each on-chip peripheral has its own specific interrupt unit containing one or more interrupt channels, or DMA channels. Please refer to the specific peripheral chapter for the description of its interrupt features and control registers.

Figure 4-13. Top Level Interrupt Structure

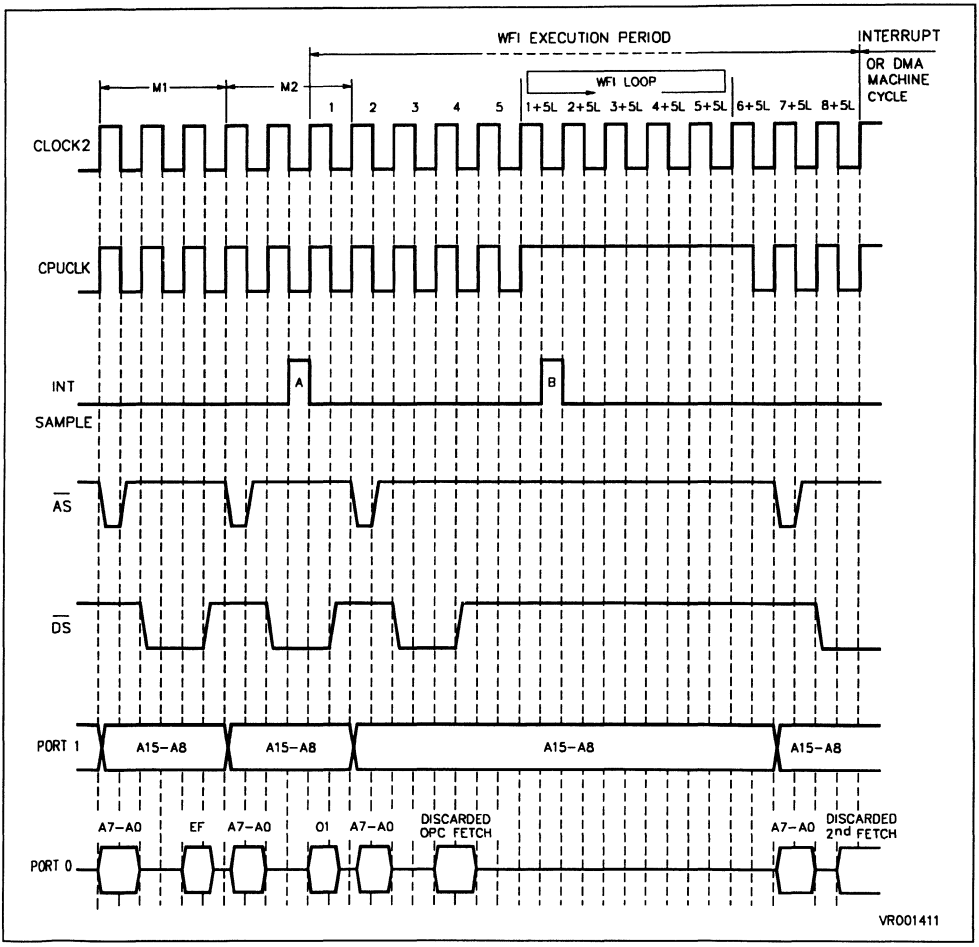


ON-CHIP PERIPHERAL INTERRUPTS (Continued)

The on-chip peripheral interrupt channels provide the following control bits:

- Interrupt Pending bit (IP)
  Set by hardware when the Trigger Event occurs. Can be set/cleared by software to generate/cancel pending interrupts and give the status for Interrupt polling.
- Interrupt Mask bit (IM)
  If IM = "0", no interrupt request is generated. If IM ="1" an interrupt request is generated whenever IP = "1" and CICR.IEN = "1".
- Priority Level (PRL, 3 bits)
  These bits define the source priority level
  PRL=0: the highest priority
  PRL=7: the lowest priority (the interrupt cannot be acknowledged)
- Interrupt Vector Register (IVR, up to 7 bits)
  The IVR points to the vector table which itself contains the interrupt routine start address.

Figure 4-14. Wait For Interrupt Timing



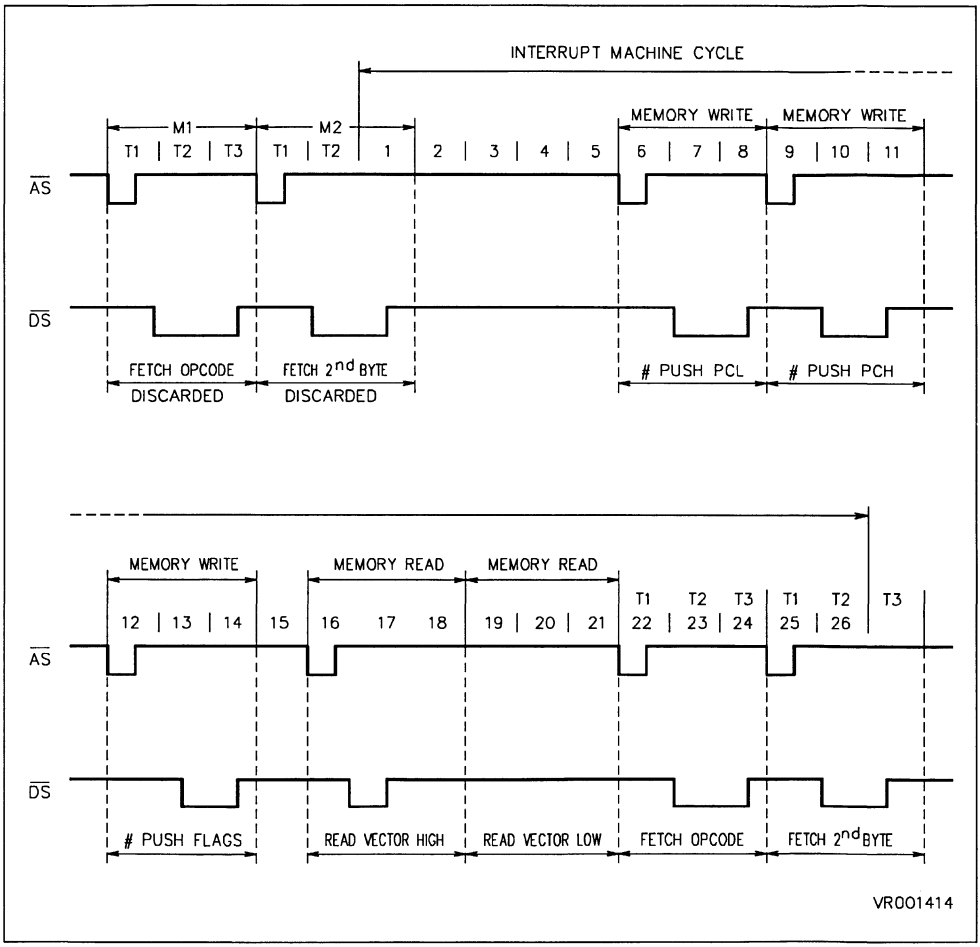
4.8 WAIT FOR INTERRUPT INSTRUCTION

The Wait For Interrupt instruction suspends program execution until an interrupt request is acknowledged. During the WFI instruction, the CPUCLK is halted while INTCLK keeps running. Under this state, the power consumption of the processor is lowered by the CORE power consumption value.

4.9 INTERRUPT RESPONSE TIME

Interrupt requests are sampled 6 CPUCLK cycles before the end of the instruction. If Wait For Interrupt is in progress, requests are sampled every 5 CPUCLK cycles. If the interrupt request comes from an external pin, the programmed event has to be set a minimum of one CPUCLK cycle before the sampling time.

Figure 4-15. Interrupt Acknowledge Timing



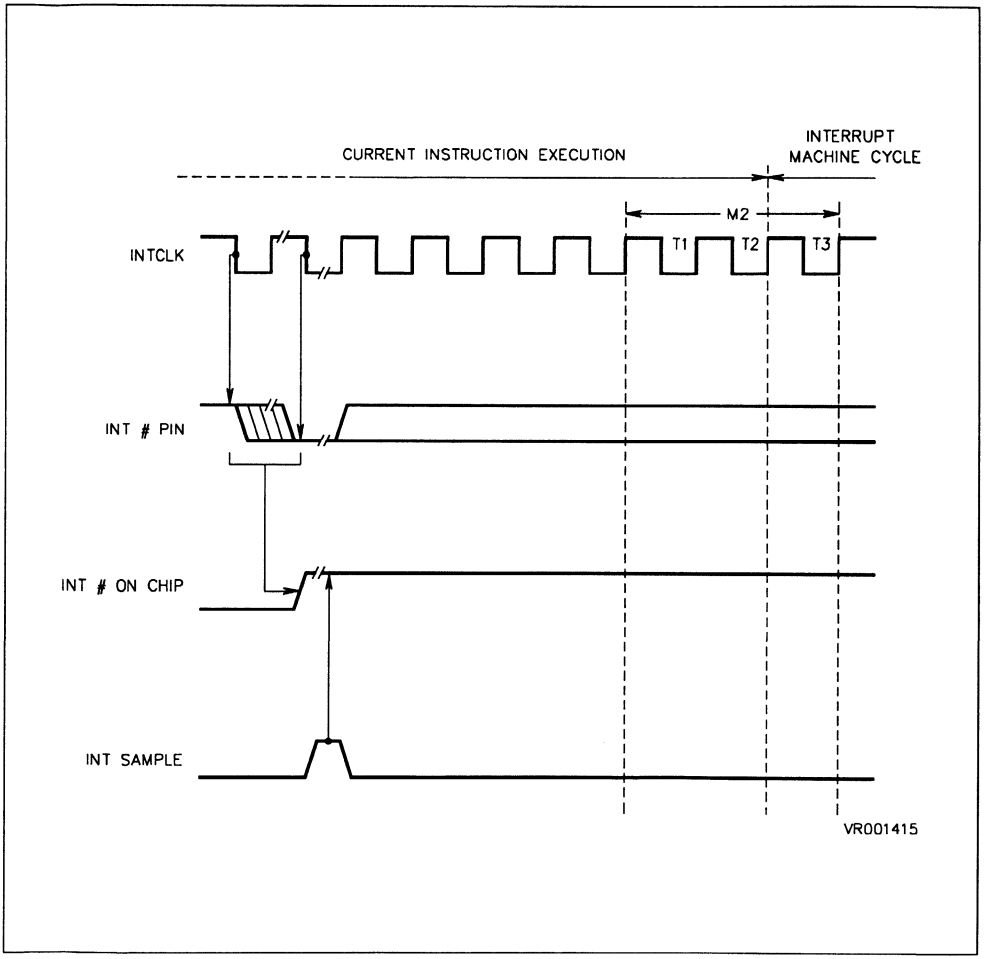
INTERRUPT RESPONSE TIME (Continued)

In order to guarantee the falling/rising edge detection, input signals must be kept low/high for a minimum of one CPUCLK cycle.

An interrupt machine cycle takes 26 internal clock cycles (CPUCLK), with some exceptions as follows:

- 28 internal clock cycles (CPUCLK), if a Wait For Interrupt is in progress
- 32 internal clock cycles (CPUCLK), if the acknowledge cycle follows a DMA transfer with Register File

Figure 4-16. External Interrupt Response Time



4.10 INTERRUPT REGISTERS

CICR R230 (E6h) System Read/Write
Central Interrupt Control Register

Reset value: 1000 0111 (87h)

7							0
GCEN	TLIP	TLI	IEN	IAM	CPL2	CPL1	CPL0

b7 = **GCEN:** *Global Counter Enable bit.* When set the 16 bit MultiFunction Timers are enabled (see Timer Control Register in MULTI FUNCTION TIMER chapter)

b6 = **TLIP:** *Top Level Interrupt Pending bit.* Set by hardware when the Trigger Event occurs. Cleared by hardware when the Top Level Interrupt is acknowledged.

b5 = **TLI:** *Top Level Interrupt bit.* If TLI ="1", and IEN is set, a Top Level Interrupt request is generated as TLIP is set. If TLI = "0", a request is generated only if TLNM is set.

b4 = **IEN:** *Interrupt Enable.* If IEN = "0", no maskable Interrupt requests are generated. This bit is cleared by the interrupt machine cycle and it is set by the IRET instruction of maskable routines.

b3 = **IAM:** *Interrupt Arbitration Mode.* If IAM = "0", Concurrent Arbitration Mode is selected; If IAM = "1" Nested Mode is selected.

b2-b0 = **CPL2, CPL1, CPL0:** *Current Priority Level.* Defines the Current Priority Level under service. CPL=0 is the highest priority. CPL=7 is the lowest priority. This bits may be modified directly by the interrupt hardware when the Nested Interrupt Mode is used.

EITR R242 (F2h) Page 0 Read/Write
External Interrupt Trigger Event Register

Reset value: 0000 0000 (00h)

7							0
TED1	TED0	TEC1	TEC0	TEB1	TEB0	TEA1	TEA0

If TExy bit is set, the pending bit will be set upon the rising edge of the input signal.

If TExy is cleared, the pending bit will be set upon the falling edge of the input signal.

All bits are set/reset only by software.

b7 = **TED1:** *Trigger Event of Interrupt Channel D1*

b6 = **TED0:** *Trigger Event of Interrupt Channel D0*

b5 = **TEC1:** *Trigger Event of Interrupt Channel C1*

b4 = **TEC0:** *Trigger Event of Interrupt Channel C0*

b3 = **TEB1:** *Trigger Event of Interrupt Channel B1*

b2 = **TEB0:** *Trigger Event of Interrupt Channel B0*

b1 = **TEA1:** *Trigger Event of Interrupt Channel A1*

b0 = **TEA0:** *Trigger Event of Interrupt Channel A0*

EIPR R243 (F3h) Page 0 Read/Write
External Interrupt Pending Register

Reset value: 0000 0000 (00h)

7							0
IPD1	IPD0	IPC1	IPC0	IPB1	IPB0	IPA1	IPA0

b7 = **IPD1:** *Interrupt Pending bit Channel D1*

b6 = **IPD0:** *Interrupt Pending bit Channel D0*

b5 = **IPC1:** *Interrupt Pending bit Channel C1*

b4 = **IPC0:** *Interrupt Pending bit Channel C0*

b3 = **IPB1:** *Interrupt Pending bit Channel B1*

b2 = **IPB0:** *Interrupt Pending bit Channel B0*

b1 = **IPA1:** *Interrupt Pending bit Channel A1*

b0 = **IPA0:** *Interrupt Pending bit Channel A0*

IP bits are hardware set upon the occurence of the trigger event and are reset by the interrupt acknowledge machine cycle.

Note. IP bits may be set by the programmer to implement a software interrupt.

INTERRUPT REGISTERS (Continued)

EIMR R244 (F4h) Page 0 Read/Write
External Interrupt Mask-bit Register

Reset value: 0000 0000 (00h)

7							0
IMD1	IMD0	IMC1	IMC0	IMB1	IMB0	IMA1	IMA0

EIMR bits are set/reset by software

When the IM bit is set (and the global IEN is enabled), an interrupt request is generated if the corresponding IP bit is set. When IM = "0", no request will be generated.

- IMxy = "1": an interrupt request can be acknowledged (depending on IEN)
- IMxy = "0": an interrupt request is masked.

b7 = **IMD1:** *Interrupt Mask of Interrupt Channel D1*

b6 = **IMD0:** *Interrupt Mask of Interrupt Channel D0*

b5 = **IMC1:** *Interrupt Mask of Interrupt Channel C1*

b4 = **IMC0:** *Interrupt Mask of Interrupt Channel C0*

b3 = **IMB1:** *Interrupt Mask of Interrupt Channel B1*

b2 = **IMB0:** *Interrupt Mask of Interrupt Channel B0*

b1 = **IMA1:** *Interrupt Mask of Interrupt Channel A1*

b0 = **IMA0:** *Interrupt Mask of Interrupt Channel A0*

EIPLR R245 (F5h) Page 0 Read/Write
External Interrupt Priority Level Register

Reset value: 1111 1111 (FFh)

7							0
PL2D	PL1D	PL2C	PL1C	PL2B	PL1B	PL2A	PL1A

EIPLR bits are set/reset by software

b7-b6 = **PL1D, PL2D:** *Priority level for the Group INTD0, INTD1*

b5-b4 = **PL1C, PL2C:** *Priority level for the Group INTC0, INTC1*

b3-b2 = **PL1B, PL2B:** *Priority level for the Group INTB0, INTB1*

b1-b0 = **PL1A, PL2A:** *Priority level for the Group INTA0, INTA1*

EIVR R246 (F6h) Page 0 Read/Write
External Interrupt Vector Register

Reset value: xxxx 0110 (X6h)

7							0
V7	V6	V5	V4	TLTEV	TLIS	IAOS	EWEN

b7-b4 = **V7 to V4:** *Most significant nibble of External Interrupt Vector.* Not initialized by reset.

b3 = **TLTEV:** *Top Level Trigger Event bit* When set, the Top Level event is triggered on rising edge of NMI input pin. Triggering on the falling edge of the NMI input pin is activated when this bit is "0" (reset value)

b2 = **TLIS:** *Top Level Input Selection bit* This bit selects the source of the Top Level Interrupt between the external NMI pin (when "1", the reset value) and the Timer/Watchdog End of Count (when "0").

b1 = **IA0S:** *Interrupt A0 Selection bit* When set, the External Interrupt pin is selected as the External Interrupt Channel A0 source. When reset the source is the Timer/Watchdog End of Count interrupt.

b0 = **EWEN:** *External Wait Enable bit.*

Must be held to the reset state.

NICR R247 (F7h) Page 0 Read/Write
Nested Interrupt Control Register

Reset value: 0000 0000 (00h)

7							0
TLNM	HL6	HL5	HL4	HL3	HL2	HL1	HL0

b7 = **TLNM:** *Top Level Not Maskable.*

If TLNM = "1", a top level request is generated as TLIP is set. Once TLNM is set, it can be cleared only with an hardware reset

bx = **HLx:** *Hold Level x* These bits are set to "1" when, in Nested Mode, an interrupt service routine at level x is interrupted from a request with higher priority (other than the Top Level interrupt request). It is cleared by the `iret` execution when the routine at level x is recovered.

NOTES

5 CLOCK

5.1 INTRODUCTION

The ST9 Clock generator module generates the internal clock for the ST9 core and the on-chip peripherals. The Clock generator can be driven by an external crystal circuit, connected to the OSCIN and OSCOUT pins, or by an external pulse generator, connected to OSCIN.

5.2 CLOCK MANAGEMENT

The oscillator circuit generates an internal clock signal with the same period and phase as at the OSCIN input pin. The maximum frequency allowed is 24MHz.

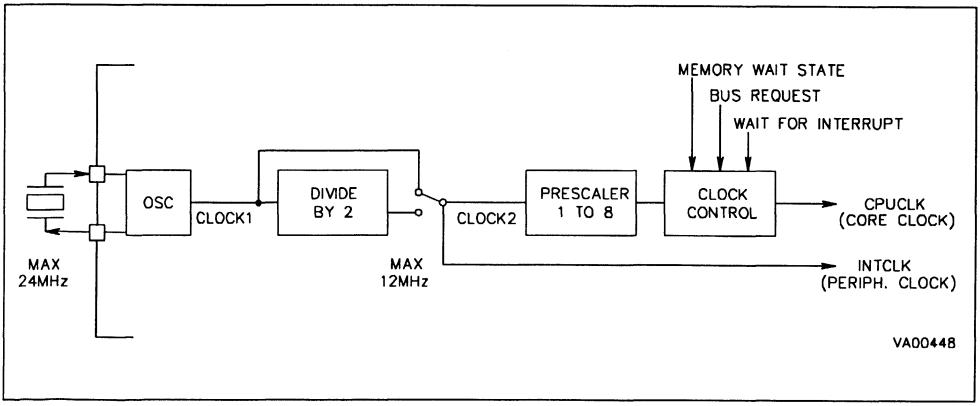
As shown in Figure 5-1, the CLOCK1 signal drives a programmable divider by two. If the control bit MODER.DIV2 (R235.5) is set, the internal clock CLOCK2 is CLOCK1 divided by two; otherwise, if DIV2 bit is cleared, the clock signal CLOCK2 has the same period and phase as CLOCK1. CLOCK2 drives the internal clock INTCLK delivered to all ST9 on-chip peripherals and acts as the central timebase for all timing functions (e.g. Multifunction Timer or Serial Communications Interface Baud Rate generator).
The maximum frequency allowed for INTCLK is 12MHz. For internal operating frequencies above 8MHz, it is recommended to work with the Clock Divider active in order to provide a duty cycle of 50% for INTCLK.

CLOCK2 also drives a programmable prescaler which generates the basic time base, CPUCLK, for the instruction executer of the ST9 core. This allows the user to slow the program execution time to reduce power dissipation, and to locally speed up certain code segments for time critical routines. The internal peripherals are not affected by the CPUCLK prescaler. The prescaler value divides the input clock by the value programmed in the control bits MODER.PRS2,1,0 (R235.4,3,2). If the prescaler value is zero, no prescale is made, thus CPUCLK has the same period and phase as CLOCK2 and INTCLK. If the value is different from 0, the prescaling is equal to the value plus one, ranging thus from two (PRS2,1,0 = 1) to eight (PRS2,1,0 = 7). The clock generated is shown in Figure 5-2. It must be noticed that the prescaling of the clock does not keep the duty cycle to 50%, but stretches the high level of the clock until completion.

When Wait for Interrupt events occur, CPUCLK is stretched on the high level for the whole period required by the function.

Note. The added wait cycles refer to the INTCLK frequency and not the original CPUCLK.

Figure 5-1. Peripheral and Core Clocks



5.3 CLOCK CONTROL REGISTER

The ST9 clock division by 2 and the clock prescaling are controlled by the MODER register.

Note. This register contains bits with other functions. Only the bits relating to control of the clock are shown here.

MODER R235 (EBh) System Read/Write Mode Register

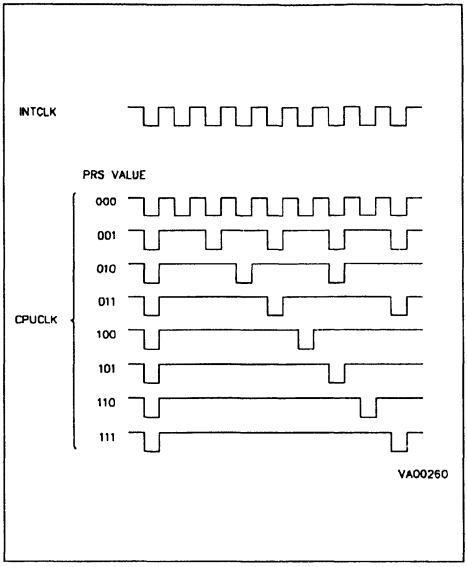
Reset Value : 1110 0000 (E0h)

7							0
X	X	DIV2	PRS2	PRS1	PRS0	X	X

b5 = **DIV2:** *OSCIN Divided by 2.* This bit controls the divide by 2 circuit which operates on the OSCIN Clock. A logical "1" value means that the OSCIN clock is internally divided by 2, and a logical "0" value means that no division of the OSCIN Clock occurs.

b4-b2 = **PRS2, PRS1, PRS0:** *Prescaling of ST9 Clock.* These bits define the prescaler value used to prescale the CPUCLK from INTCLK. When these three bits are reset, the CPUCLK is not prescaled, and is equal to INTCLK; in all other cases, the internal clock is prescaled by the value of (PRS2,1,0 + 1).

Figure 5-2. Core Clock Prescaling



5.4 OSCILLATOR CHARACTERISTICS

The on-chip oscillator circuit (Figure 5-3) is an inverting gate circuit.

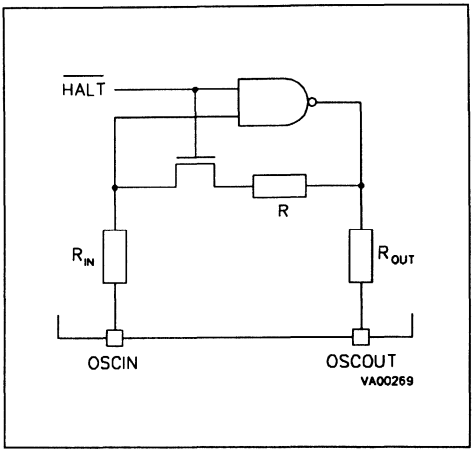
Note. Owing to the Q factor required, Ceramic Resonators may not provide a reliable oscillator source.

In Halt mode, set by means of the HALT instruction, the parallel resistor R is disconnected and the oscillator is disabled, forcing the internal clock CLOCK1 to a high level and OSCOUT to a high level.

To exit the HALT condition and restart the oscillator, an external RESET pulse is required of a minimum duration of 12ms (Figure 5-6).

It must be noted that if the Timer/Watchdog watchdog function is enabled, a HALT instruction will not disable the oscillator. This to avoid stopping the watchdog if, by an error, a HALT code is executed. When this occurs, the ST9 CPU falls into an endless loop ended by the watchdog (or external) reset.

Figure 5-3. Internal Oscillator Schematic



Note: $R_{OUT} < 50\Omega$ $R > 1M\Omega$ $300\Omega < R_{IN} < 500\Omega$

Figure 5-4. Crystal Oscillator

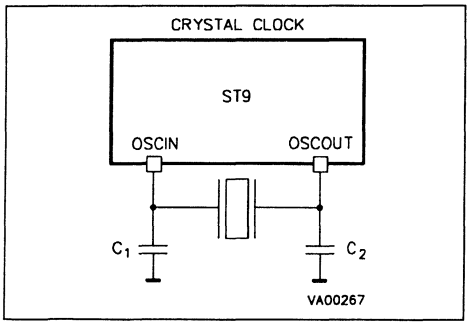


Figure 5-5. External Clock

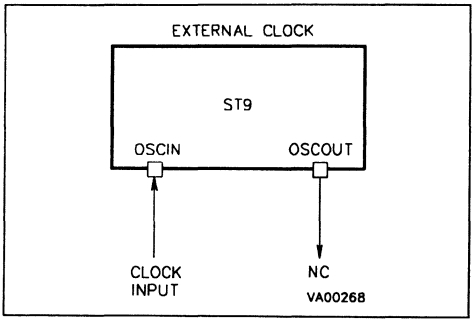


Table 5-1. Crystal Specification (C0 <= 7pF)

Frequency (MHz)	C1=C2=56pF Rs Max	C1=C2=47pF Rs Max	C1=C2=22pF Rs Max
24	20	25	70
16	40	60	150
12	80	100	250
8	180	240	600
4	700	800	600

Table 5-2. Oscillator Transductance

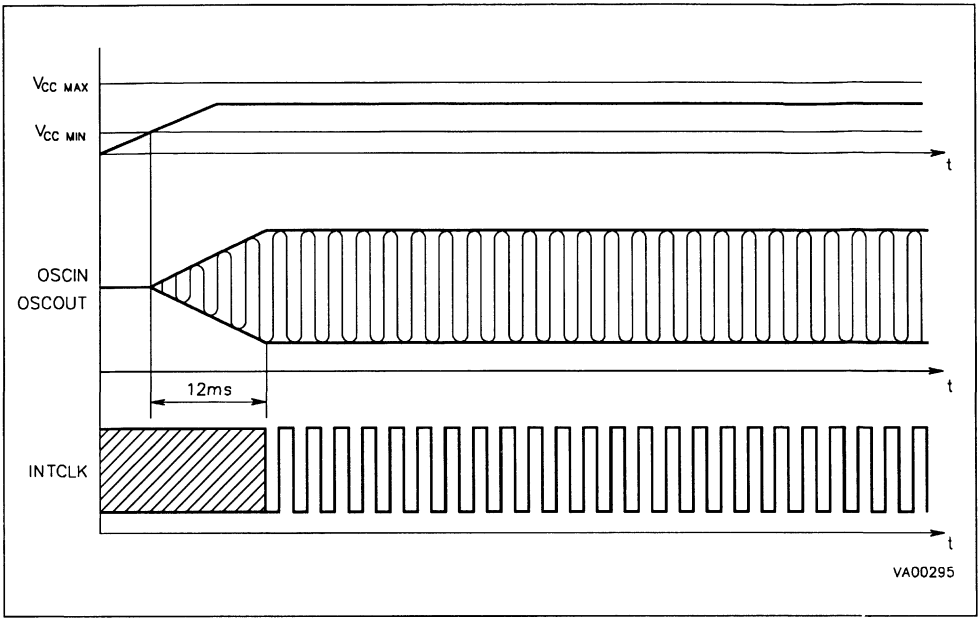
gm	Min	Typ	Max
mA/V	3	5.8	9.5

Legend:
Rs: Parasitic Series Resistance of the quartz crystal (upper limit)
C0: Parasitic capacitance of the quartz crystal (upper limit, < 7pF)
C1, C2: Maximum Total Capacitances on pins OSCIN and OSCOUT(the value includes the external capacitance tied to the pin plus the parasitic capacitance of the board and of the device)
gm: Transconductance of the oscillator

Note. The tables are relative to the fundamental quartz crystal only (not ceramic resonator).

OSCILLATOR CHARACTERISTICS (Continued)

Figure 5-6. Oscillator Start-up Sequence

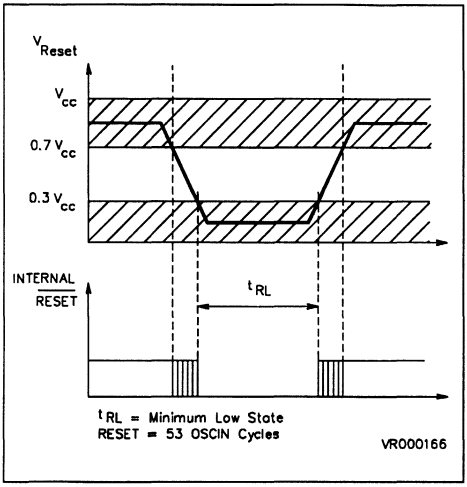


6 RESET

6.1 INTRODUCTION

The processor Reset overrides all the other conditions and forces the ST9 to the reset state. During Reset, the internal registers are set to the reset value, as shown in Table 6-1 for the system and Page 0 Registers and the I/O pins are set to the Bidirectional Weak Pull-up mode (see Warning). The programmer must then initialize the ST9 system and peripheral control registers to give the required functions.

Figure 6-1. Signal to be applied on Reset Pin



6.2 RESET GENERATION

The reset condition can be generated by the external pin $\overline{\text{RESET}}$ or by the on-chip Timer/Watchdog.

The on-chip Timer/Watchdog generates a reset condition if the watchdog mode is enabled (WCR.WDEN cleared, R252 page 0), and if the programmed period elapses without the specific code (AAh,55h) written to the appropriate register. The input pin RESET is not driven low by the on-chip reset generated by the Timer/Watchdog.

6.3 RESET PIN TIMING

The $\overline{\text{RESET}}$ pin has a Schmitt trigger input circuit with hysteresis. The internal reset is generated by the external pin synchronized with the internal clock. The power up reset circuit must keep the $\overline{\text{RESET}}$ input low for a minimum of the crystal startup period plus 53 crystal periods.

Once the $\overline{\text{RESET}}$ pin reaches a logical "1", the processor exits from the reset status after 67 crystal periods from the rising edge. The processor then fetches from Program Memory locations 0 and 1 (power-on reset vector) and begins program execution from the address contained in the vector.

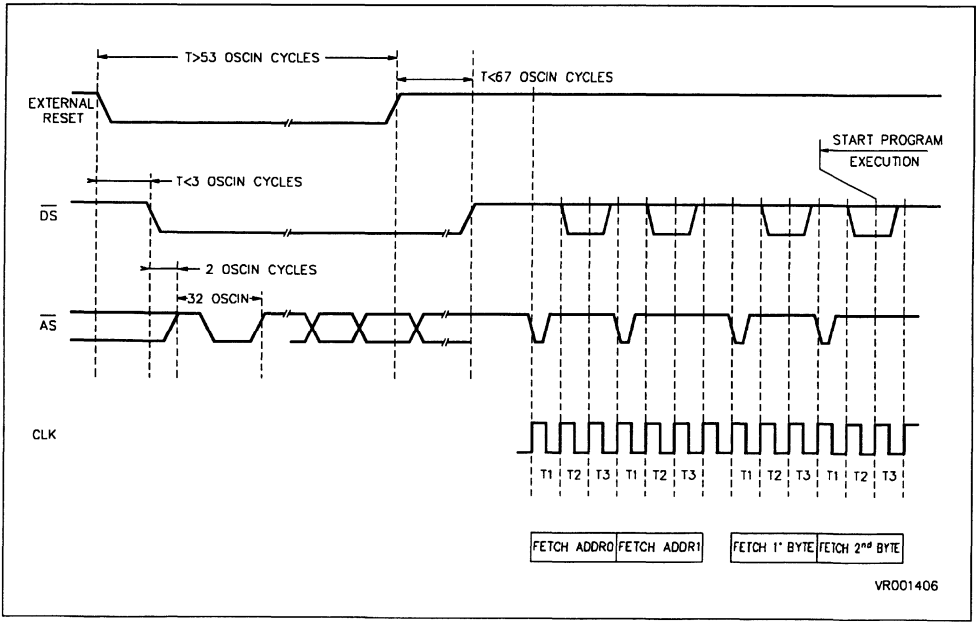
WARNING: *I/O pins are set to the Weak Pull-up mode during the Reset cycle. This state is forced during the reset sequence, but the I/O pins can be in a random state for up to 64 crystal periods. The application circuit must take this into account if it can lead to critical situations in the external circuitry.*

RESET (Continued)

Table 6-1. Internal Registers Reset Values

Register Number	System Register	Reset Value	Page 0 Register	Reset Value
F	(SSPLR)	undefined	(SWAP)	xxx0 xxxxb
E	(SSPHR)	undefined	(SPICR)	00h
D	(USPLR)	undefined	(SPIDR)	undefined
C	(USPHR)	undefined	(WCR)	7Fh
B	(MODER)	E0h	(WDTCR)	12h
A	(Page Ptr)	undefined	(WDTPR)	undefined
9	(Reg Ptr 1)	undefined	(WDTLR)	undefined
8	(Reg Ptr 0)	undefined	(WDTHR)	undefined
7	(FLAGR)	undefined	(NICR)	00h
6	(CICR)	87h	(EIVR)	x2h
5	(PORT5)	FFh	(EIPLR)	FFh
4	(PORT4)	FFh	(EIMR)	00h
3	(PORT3)	FFh	(EIPR)	00h
2	(PORT2)	FFh	(EITR)	00h
1	(PORT1)	FFh	Reserved	
0	(PORT0)	FFh	Reserved	

Figure 6-2. Exit From Reset Timing



7 I/O PORTS

7.1 INTRODUCTION

The ST9 is provided with dedicated lines for input/output. These lines, grouped into 8-bit ports, can be independently programmed to provide parallel input/output, or to carry input/output signals to/from the on-chip peripherals and Core (e.g. SPI and OSD). All ports have active pull-ups and pull-down resistors compatible with TTL loads. In addition, pull-ups can be turned off for open-drain operation and weak pull-ups can be turned on to save off-chip resistive pull-ups. Input buffers can be either TTL or CMOS compatible.

7.2 CONTROL REGISTERS

Each port PX has a Data Register PX, and three associated control registers (PXC0, PXC1, PXC2) which define the port line configuration and allow dynamic change in port configuration during program execution. Ports and control registers are mapped into the Register File as shown in Figure 7-1 Ports and control registers are treated like any other general-purpose register. There are no special instruction for port manipulation, any instruction that addresses a register can address the ports. Data can be directly accessed in the port register, without passing through other memory or "accumulator" locations.

Figure 7-1. I/O Register Maps

Applicable to ST9294

GROUP E			GROUP F		
				PAGE 2	PAGE3
			FFh	Reserved	Reserved
			FEh	P3C2	Reserved
			FDh	P3C1	Reserved
			FCh	P3C0	Reserved
			FBh	Reserved	Reserved
			FAh	P2C2	Reserved
			F9h	P2C1	Reserved
			F8h	P2C0	Reserved
			F7h	Reserved	Reserved
			F6h	P1C2	P5C2
E5h	P5DR	R229	F5h	P1C1	P5C1
E4h	P4DR	R228	F4h	P1C0	P5C0
E3h	P3DR	R227	F3h	Reserved	Reserved
E2h	P2DR	R226	F2h	P0C2	P4C2
E1h	P1DR	R225	F1h	P0C1	P4C1
E0h	P0DR	R224	F0h	P0C0	P4C0

CONTROL REGISTERS (Continued)

During the reset state, all the Ports are set as bidirectional/weak pull-up mode with the output data register set to FFh. This condition is also held after reset (see Warning on page 50) and can be redefined under software control at any time.

7.3 PORT BIT STRUCTURE AND PROGRAMMING

By programming the control bits PXC0.n and PXC1.n (see Figure 7-2) it is possible to configure bit PX.n as Input, Output, Bidirectional or Alternate Function Output, where X is the number of the I/O port, and n the bit within the port (n = 0 to 7).

When programmed as an Input, the pin may be set to TTL or CMOS input threshold levels, except for P2.0 and P2.1, which are implemented with a Schmitt trigger function. Any input pin can be set to the Schmitt trigger function by ROM mask option.

For a bit programmed as Output, the Push-Pull or Open Drain configuration may be selected. The bidirectional mode sets the pin to Open Drain, allowing bidirectional communication with external logic. Certain pins can also be set with a weak pull-up resistor to V_{DD} in bidirectional mode (see next paragraph).

Warning: *All pins not externally available on the 42-pin package, but present on the 56-pin package (P1.0 - P1.7, P3.7, P5.2 and P5.3) have the weak pull-up bidirectional mode. The reset configuration for these pins is bidirectional mode with weak pull-up.*
All other pins are reset to the bidirectional mode with weak pull-up BUT with the weak pull-up disabled. This gives the behaviour of Bidirectional Open Drain Mode.

When configured as Open Drain, the voltage on the pin must never exceed the V_{DD} power line value. Port 4 has no push-pull or weak pull-up driving capabilities but has a true Open Drain feature up to 12 volts.

The basic structure of the bit PX.n of a general purpose port PX is shown in Figure 7-3.

Figure 7-2. Control Bits

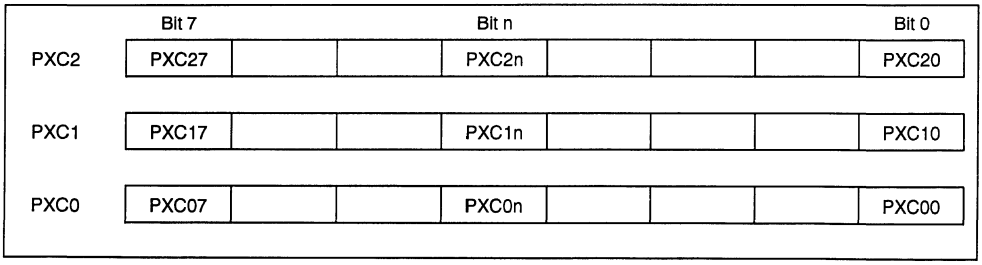


Table 7-1. Port Bit Configuration Table

PXC2n PXC1n PXC0n	0 0 0	1 0 0	0 1 0	1 1 0	0 0 1	1 0 1	0 1 1	1 1 1	
PXn Configuration	BID	BID	OUT	OUT	IN	IN	AF	AF	
PXn Output	WP	OD	PP	OD	HI	HI	PP	OD	HI(1)
PXn Input	TTL	TTL	TTL	TTL	CMOS	TTL	TTL	TTL	V_{SS}(1)

Legend:
X = Port
n = Bit
AF = Alternate Function
HI = High Impedance
OD = Open Drain
WP = Weak Pull-up
PP = Push-Pull
BID = Bidirectional
TTL = TTL Standard Input
CMOS = CMOS Standard Input
IN = Input
OUT = Output

Note 1: For A/D Converter inputs

PORT BIT STRUCTURE AND PROGRAMMING (Continued)

Figure 7-3. Basic Structure of an I/O Port Pin

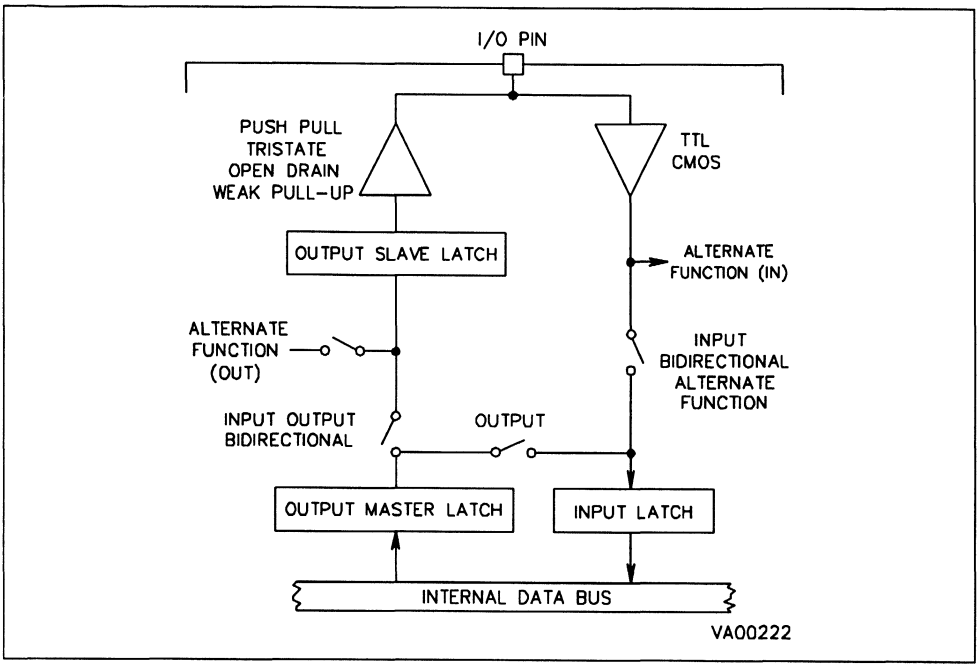


Figure 7-4. Input Configuration

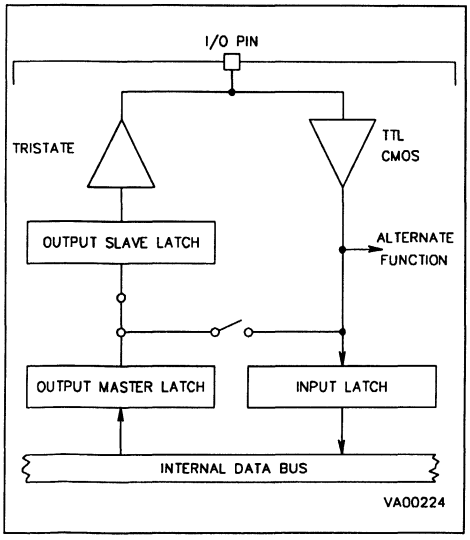
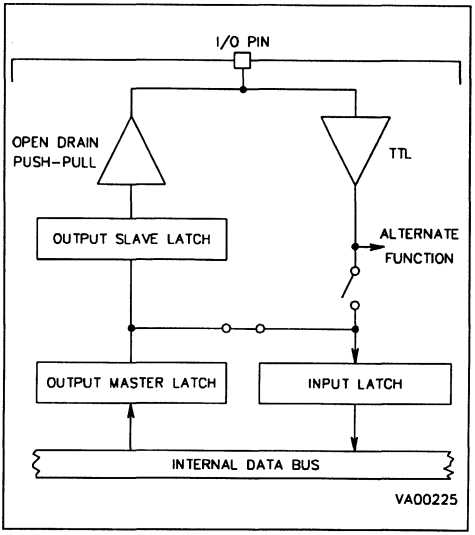


Figure 7-5. Output Configuration



PORT BIT STRUCTURE AND PROGRAMMING (Continued)

Independently to the chosen configuration, when the User addresses the port as an destination register of an instruction, the port is written to and the data is transferred from the internal Data Bus into the Output Master Latches. When the port is addressed as a source register for an instruction, the port is read and the data stored in the Input Latch is transferred onto the internal Data Bus.

When PX.n is programmed as Input: (Figure 7-4)

- The Output Buffer is forced tristate
- The data present on the I/O pin is sampled into the Input Latch at the beginning of the execution of the instruction which is accessing the port.
- The data stored in the Output Master Latch is copied into the Output Slave Latch at the end of the execution of each instruction. So if bit PX.n is reconfigured as Output or Bidirectional, the data stored in the Output Slave Latch is reflected on the I/O pin.

When PX.n is programmed as Output: (Figure 7.5)

- The Output Buffer is turned on in an Open-drain or Push-pull configuration
- The data stored in the Output Master Latch is copied both into the Input Latch and into the Output Slave Latch, driving the I/O pin, at the end of the execution of each instruction.

When PX.n is programmed as Bidirectional: (Figure 7-6)

- The Output Buffer is turned on in an Open-drain or Weak Pull-up configuration
- The data present on the I/O pin is sampled into the Input Latch at the beginning of the execution of each instruction
- The data stored in the Output Master Latch is copied into the Output Slave Latch, driving the I/O pin, at the end of the execution of each instruction.

WARNING. Due to the unique feature of the bidirectional mode of reading the external pin instead of the output latch, particular care must be taken with arithmetic/logic and boolean instructions performed on a bidirectional port pin.

These instructions use a read-modify-write sequence, and the result written in the port register depends on the logical level present on the external pin.

This may bring unwanted modifications to the port output register content.

For example:

Port register content	external port value
0Fh	03h

(Bits 3 and 2 are externally forced to 0)

Making a `bset` instruction on bit 7 will return:

Port register content	external port value
83h	83h

(Bits 3 and 2 have been cleared.)

Figure 7-6. Bidirectional Configuration

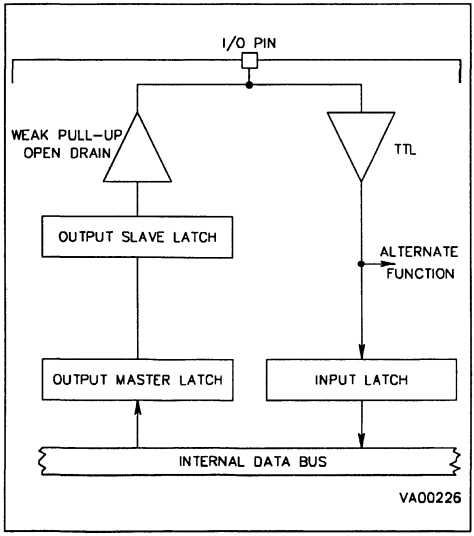
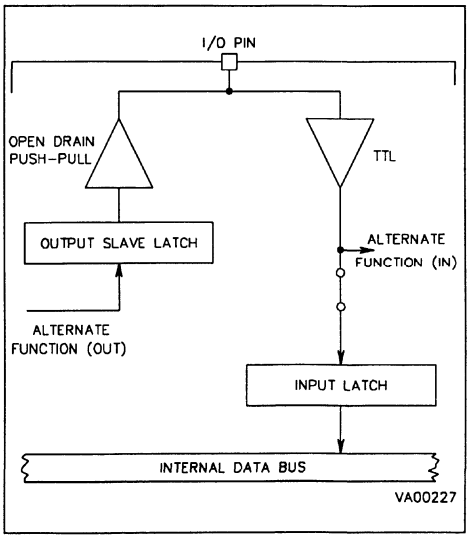


Figure 7-7. Alternate Function Configuration



PORT BIT STRUCTURE AND PROGRAMMING (Continued)

To avoid this situation, it is suggested that all the operations on a port, using at least one bit in bidirectional mode, are performed on a copy of the port register, then transferring the result with a load instruction to the I/O port.

When PX.n is programmed as Alternate Function Output (Figure 7-7) except for Analog Inputs :

- The Output Buffer is turned on in an Open-drain or Push-pull configuration
- The data present on the I/O pin is sampled into the Input Latch at the beginning of the execution of each instruction
- A signal coming from an on-chip Function is allowed to load the Output Slave Latch driving the I/O pin. Signal timing is under control of the Function. If no Function is connected to PX.n the I/O pin is driven to a high level in Push-pull configuration and is driven to high impedance in open drain configuration.

7.4 ALTERNATE FUNCTION ARCHITECTURE

Each single I/O pin may access three different types of ST9 internal signals:

- Data bus line (I/O)
- 'Alternate Function' Input
- Alternate Function Output

Each pin configuration is made by software, thus allowing the User to choose the type of signal to access a pin. The choice of type of signal is made with the registers PXC2, PXC1, PXC0 of the I/O Port X (Please refer to the previous section for more details)

Pins Declared as an I/O
A pin declared as an I/O is a pin connected to the I/O buffer. In such a case, this pin may either be an Input or an Output or an I/O depending on the value stored in (PXC2, PXC1, PXC0)

Pin Declared As An 'Alternate Function' Input
A single pin may be directly connected to several Alternate Function inputs. In such a case, the User has to select the required input mode (TTL or

Figure 7-8. Example of 3 Alternate Function Inputs

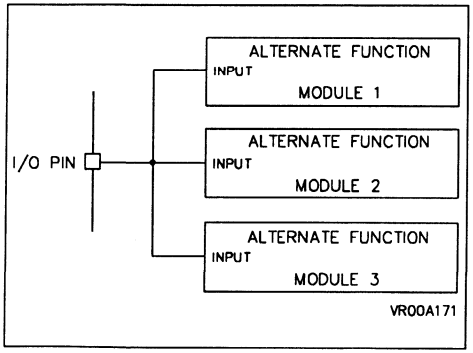
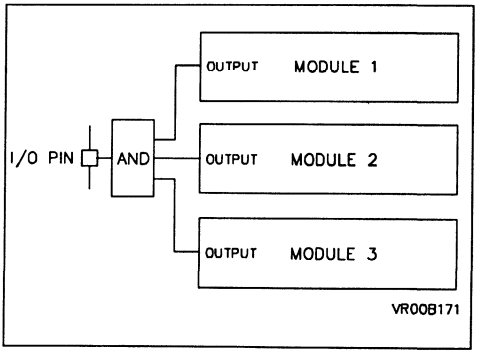


Figure 7-9. Example of 3 Alternate Function Inputs



ALTERNATE FUNCTION (Continued)

CMOS levels) and to enable, by software, the selected Alternate Function module (by enabling it) and unselect all other Alternate Functions (by disabling them).
No specific configuration of the port is required to enable the input Alternate Function, as the input buffer is directly connected to each module using it. As more than one module can use the same input Alternate Function line, it is under User software control to enable a module to use the input Alternate Function.
The digital I/O remains operational even when using the Alternate Function input. The exception to this is for an I/O port bit connected to analog voltages (for the Analog to Digital Converter, see Figure 9.10).

Pin Declared As An Alternate Function Output
A pin declared as an Alternate function output corresponds to (PXC2,PXC1,PXC0) = 1,1,1 or 0,1,1. Several Alternate Function outputs may drive a common pin. In such a case, the Alternate Function output signals are ANDed before driving the common pin. The User has therefore to select, by software, the Alternate Function Output required by enabling it and disabling all other Alternate Function Outputs on the same pin (a disabled Alternate Function Output outputs a "1").

The inputs to on-chip Functions and Alternate Function Outputs are predefined for each I/O pin. Please refer to the Alternate Function Table at the beginning of this datasheet for the exact configuration.

WARNING: *The user must take care not to program I/O pins present in the 56 pin package, but not available in the 42 pin package, to the input tristate mode. This is to avoid spurious and extra power consumption that may appear due to CMOS floating inputs.*

General Configuration
A single pin may be used, according to different phases of the software, as an I/O or connected to an input or an Alternate Function output.

WARNING: *When a pin is connected to an Input Function and to an Alternate Function output, the User must be aware of the fact that the Alternate Function output signal always input to the Alternate Function module(s) declared as input(s).*

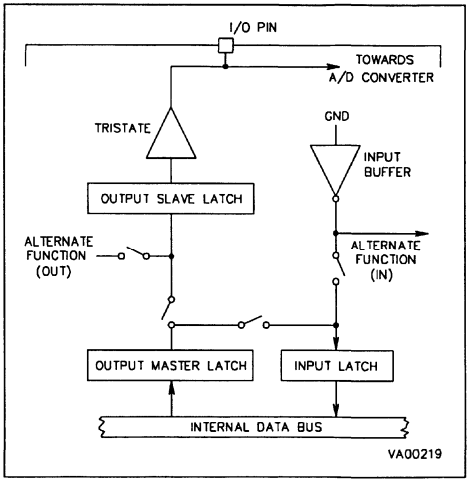
7.5 SPECIAL PORTS

7.5.1 Bit Structure for A/D Converter Inputs

When a port bit is used as input for an on-ship A/D Converter, its structure is modified as shown in Figure

The behaviour of this bit is identical to the general purpose bit described in paragraph 9.68 except when it is programmed as Alternate Function. In this case, the Output Buffer is forced Tristate and the input of the Input Buffer is disconnected from the I/O pin and forced low. In this way the I/O pin is free to assume any analog value without causing power consumption in the Input Buffer. The bit **MUST** be programmed to **(PXC2, PXC1, PXC0)=1,1,1)** to assume this special configuration.

Figure 7-10. A/D Input Port Bit Structure



7.6 I/O STATUS AFTER WFI, HALT AND RESET

The status of the ST9 I/O ports during the Wait For Interrupt, Halt and Reset operational modes is set to the bidirectional weak pull-up state, except for P3.3 and P3.4 which have no weak pull-up.

WARNING : *I/O pins are set to the Weak Pull-up mode during the Reset cycle. This state is forced during the reset sequence, but the I/O pins can be in a random state for up to 64 crystal periods.*
The application circuit must take this into account if it can lead to critical situations in the external circuitry.

8 SERIAL PERIPHERAL INTERFACE

8.1 INTRODUCTION

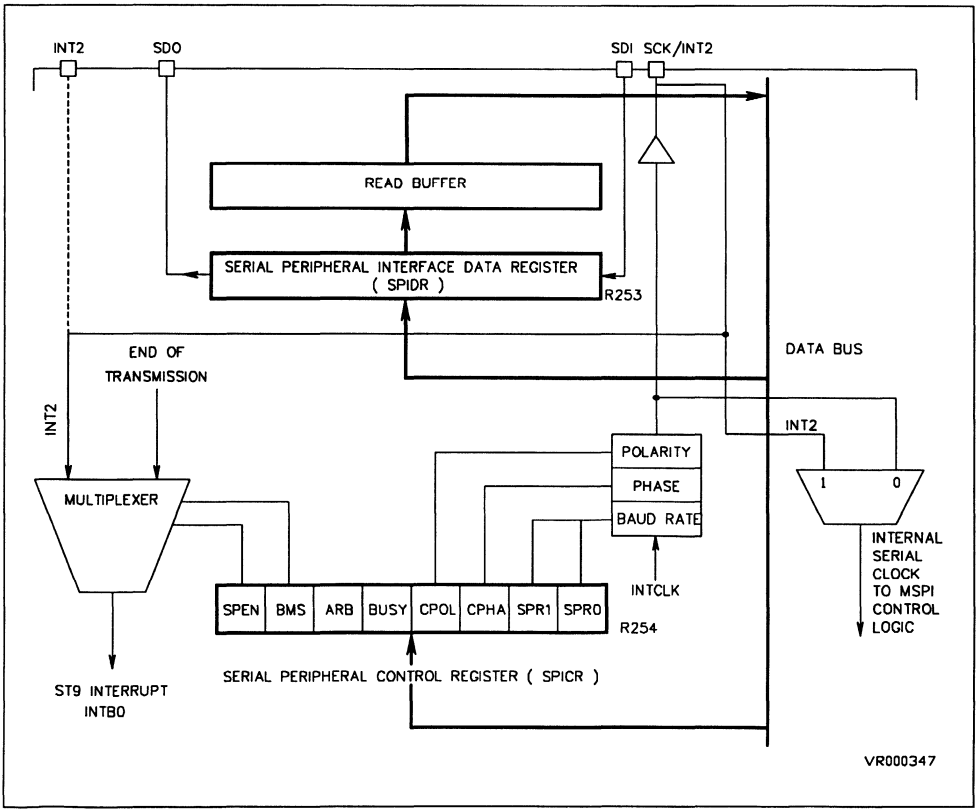
The Serial Peripheral Interface (SPI) is integrated into the Core module of the ST9 and provides a general purpose shift register based peripheral allowing several external peripherals to be linked through an SPI protocol bus. In addition, special modes allow reduced software overhead with I^2C-bus and IM-bus Communication standards.

The SPI uses 3 lines comprising Serial Data In (SDI) and Alternate Function outputs Serial Data Out (SDO) and Synchronous Serial Clock (SCK). Additional I/O pins may act as device selects or IM-bus address ident signals.

Its Main Features are:

- Full duplex 3-wire synchronous transfer
- Master operation only
- 1.5MHz max bit transfer frequency (INTCLK = 12MHz)
- 4 Programmable bit rates
- Programmable clock polarity and phase
- Busy Flag
- End of transmission interrupt
- Additional hardware to facilitate more complex protocols

Figure 8-1. Block Diagram



8.2 FUNCTIONAL DESCRIPTION

The SPI, when enabled, receives input data from the ST9 Core internal data bus into SPIDR, and originates the Serial Clock (SCK) based upon dividing of the internal processor clock (INTCLK). The data is parallel loaded into the 8 bit shift register (from the internal bus) during a write cycle and then shifted out serially through the SDO pin (Most Significant bit first) to the slave device, which responds by sending its data to the master device via the SDI pin. This implies full duplex transmission with data-out and data-in both synchronized with the same clock signal. Thus the transmitted byte is replaced by the byte received, eliminating the need to have separate "Tx empty" and "Rx full" status bits.

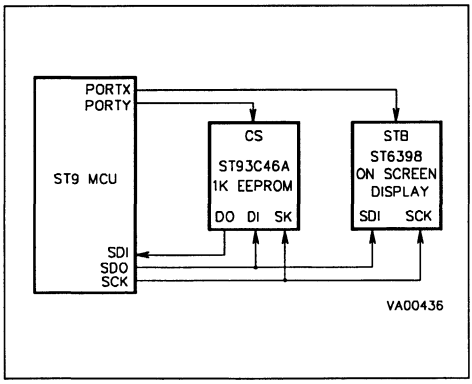
When the shift register is loaded, data is parallel transferred to the read buffer and data becomes available for the ST9 during a following read cycle.

The SPI requires three pins on an I/O port:

SCK	Serial Clock signal
SDO	Serial Data Out
SDI	Serial Data In

An additional output bit of an I/O port may be used to perform the slave chip select signal.

Figure 8-2. A Typical SPI Network



8.2.1 Input Signal Description

Serial Data In (SDI)

Data is transferred serially from a slave to a master on this line, most significant bit first. In an S-BUS/I^2C-bus configuration, SDI line senses the value forced on the data line (by SDO or by another peripheral connected to the S-bus/I^2C-bus environment).

8.2.2 Output Signal Description

Serial Data Out (SDO)

The SDO pin is configured as an output for the master device. This is obtained by programming the corresponding I/O pin as an output alternate function. Data is transferred serially from a master to a slave on SDO, most significant bit first. This pin is forced to the high impedance state when the SPI is disabled and is set to "1" when arbitration is lost (during an S-bus/I^2C-bus protocol transmission). The master device always allows data to be applied on the SDO line one half cycle before the clock edge in order to latch the data for the slave device.

Master Serial Clock (SCK)

The master device uses SCK to latch the incoming data on the SDI line. This pin is forced to a high impedance state when SPI is disabled (SPEN, SPICR.7 = "0"), in order to avoid clock contention from different masters in a multi-master system.
The master device generates SCK from INTCLK. SCK is used to synchronize the transfer of data both in and out of the device through its SDI and SDO pins. The SCK type and its relationship to data are controlled by the CPOL and CPHA bits in the Serial Peripheral Control Register.
This input is provided with a digital filter which cleans spikes lasting less than one INTCLK period.

Two bits (SPR1 and SPR0) in the Serial Peripheral Control Register, SPICR (R254) select the clock rate. Four frequencies can be selected, two in a high frequency range (mostly used with the SPI protocol) and two in a medium frequency range (mostly used for more complex protocols).

8.3 INTERRUPT STRUCTURE

SPI peripheral is associated with external interrupt channel B0 (pin INT2). Multiplexing between the external pin and SPI internal source is controlled by the SPEN and BMS bits according to the following table.

The two possible SPI interrupt sources are: End of transmission (after each byte) and S-bus/I²C-bus start condition. Care should be taken when toggling SPEN or/and BMS bits from (0,0) status, this should be done by masking the interrupt channel B0 (reset of EIMR.IMB0, bit 2 of External Interrupt Mask Register). Furthermore it is necessary to clear possible spurious requests on the corresponding channel by resetting the interrupt pending bit EIPR.IPB0 (bit 2 of External Interrupts Pending Register).

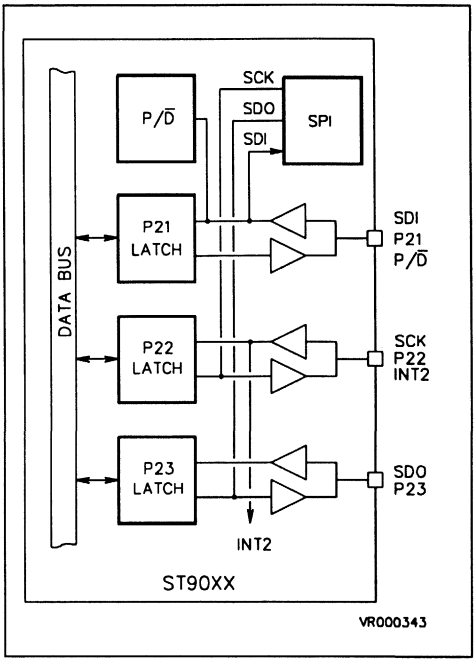
The INT2 input Function is always mapped together with the SCK input Function to allow start/stop bit detection when using S-bus/I²C-bus protocols.

A delay instruction (e.g. a NOP instruction) should be inserted between the SPEN toggle instruction and the interrupt pending bit reset instruction.

Table 8-1. Interrupt Configuration

SPEN	BMS	Interrupt Source
0	0	External channel INT2
0	1	S-bus/I²C bus start or stop condition
1	X	End of one byte transmission

Figure 8-3. SPI I/O Pins



8.4 SPI REGISTERS

SPI uses three registers mapped on page 0 of the register file:

SPIDR R253 (FDh) Page 0 Read/Write
SPI Data F egister

Reset Value: 0000 0000b (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

b7-b0 = **D0-D7:** *SPI Data Bits.* This register contains the data transmitted and received by the SPI. Data is transmitted b7 first, and receives incoming data into b0. Transmission is started by writing to this register.

SPICR R254 (FEh) Page 0 Read/Write
SPI Control Register

Reset Value: 0000 0000b (00h)

7							0
SPEN	BMS	ARB	BUSY	CPOL	CPHA	SPR1	SPR0

b7 = **SPEN:** *Serial Peripheral Enable.* When set, the two alternate functions SCK and SDO are enabled. When disabled, SCK and SDO are kept in high impedance. Furthermore, SPEN affects the selection of the source for interrupt channel B0. Transmission will start by simply writing the data into the SPIDR Register.

b6 = **BMS:** *S-bus/I²C-bus Mode Selector.* This bit should be set to "1" when the SPI is used in an S-bus/I²C-bus protocol. It enables S-bus/I²C-bus arbitration, clock synchronization and Start/ Stop detection.
When this bit is reset to "0", a reinitialisation of the SPI logic is performed allowing recovery procedures after a Rx/Tx failure. BMS (and SPEN) affects the selection of the source for interrupt channel B0.

b5 = **ARB:** *Arbitration flag bit.* This bit is set when the SPI, in S-bus/I²C-bus mode, loses arbitration, and is reset when an S-bus/I²C-bus stop condition is detected. ARB can be reset by software. When ARB is set automatically, the SDO pin is set to high value until a write instruction on SPIDR is performed.

b4 = **BUSY:** *SPI Busy Flag.* BUSY flag is set when a transmission is in process. This bit allows the user to monitor the SPI status by polling its value.

b3 = **CPOL:** *Transmission Clock Polarity.* CPOL controls the normal or steady state value of the clock when data is not being transferred.

As the SCK line is held in a high impedance state when the SPI is disabled (SPEN = "0"), the SCK pin must be connected to V_{SS} or V_{CC} through a resistor according to the CPOL state. Polarity should be selected during the reset routine according to the value set into all peripherals and must not be changed during program execution.

b2 = **CPHA:** *Transmission Clock Phase.* CPHA controls the relationship between the data on the SDI and SDO pins and the clock produced at the SCK pin. CPHA bit selects the clock edge which captures data and allows it to change state. It has its greatest impact on the first bit transmitted (MSB) because it does (or does not) allow a clock transition before the first data capture edge.
Figure 8-5 shows the relationship between CPHA, CPOL and SCK, and indicates active clock edges and strobe times.

CPOL	CPHA	SCK on Figure 8-5
0	0	(a)
0	1	(b)
1	0	(c)
1	1	(d)

b1-b0 = **SPR1,SPR0:** *SPI Rate.* These two bits select one (out of four) baud rates to be used as SCK.

SPR1	SPR0	Clock Divider	SCK Frequency (INTCLK = 12MHz)	
0	0	8	1500kHz	(T = 0.67µs)
0	1	16	750kHz	(T = 1.33µs)
1	0	128	93.75kHz	(T = 10.66µs)
1	1	256	46.87kHz	(T = 21.33µs)

SWAP R255 (FFh) Page 0 Read/Write
SPI SWAP Control Register

Reset Value: XXX0 XXXX

7							0
X	X	X	SWAP	X	X	X	X

b4 = *SWAP bit*: The Swap bit allows the SDIO and SCK/INT2 functions to be swapped in order to allow circuit flexibility.

SPI SWAP Function

SWAP	P5.0	P5.1
0	SCK/INT2	SDIO
1	SDIO	SCK/INT2

8.5 WORKING with DIFFERENT PROTOCOLS

The SPI peripheral offers the following facilities to work with S-bus/I²C-bus and IM-bus protocols:

- Interrupt request on start/stop detection
- Hardware clock synchronisation
- Arbitration lost flag with an automatic set of data line

Note that the I/O bit associated to the SPI should be returned to a defined state as a normal I/O pin before changing the SPI protocol.

The following paragraphs provide information to manage these protocols.

8.5.1 I²C-bus Interface

I²C-bus is a two-wire bidirectional data-bus, the two lines being SDA (Serial DAta) and SCL (Serial CLock). Both are open drain lines to allow arbitration. As shown in figure 8-6, data is toggled with clock low and Start and Stop conditions are detected when a high to low (start) or a low to high (stop) transition on the SDA line occurs with the SCL line high.

Each transmission consists of nine clock pulses (SCL line). The first 8 pulses transmit the byte (msb first), the ninth is used by the receiver to acknowledge.

Figure 8-4. S-Bus/I²C-bus Peripheral Compatibility without S-Bus Chip Select

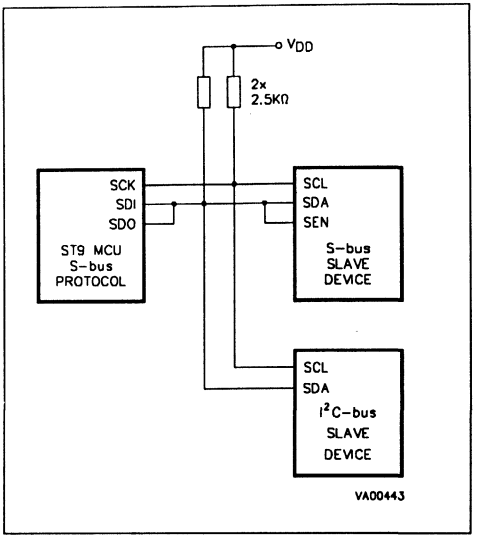
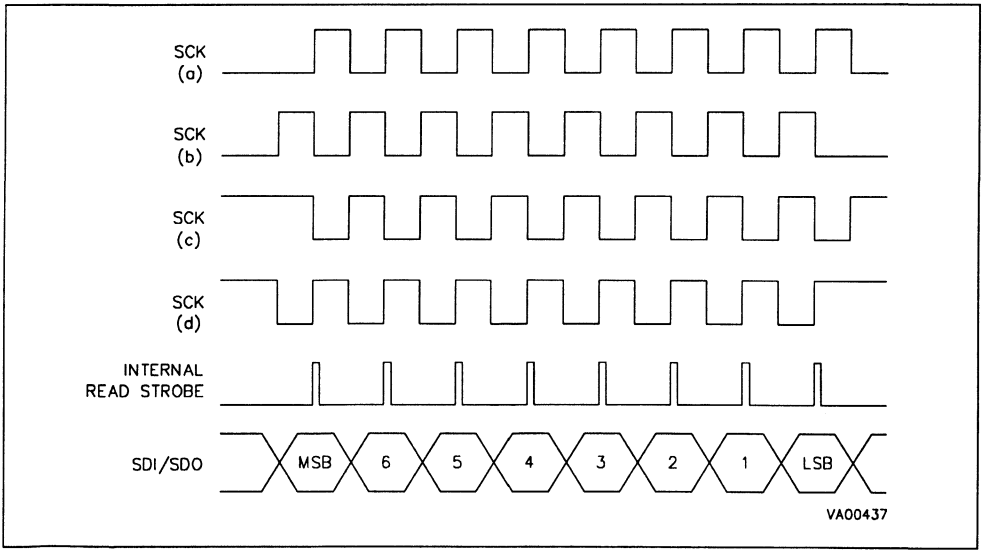


Figure 8-5. SPI Data and Clock Timing

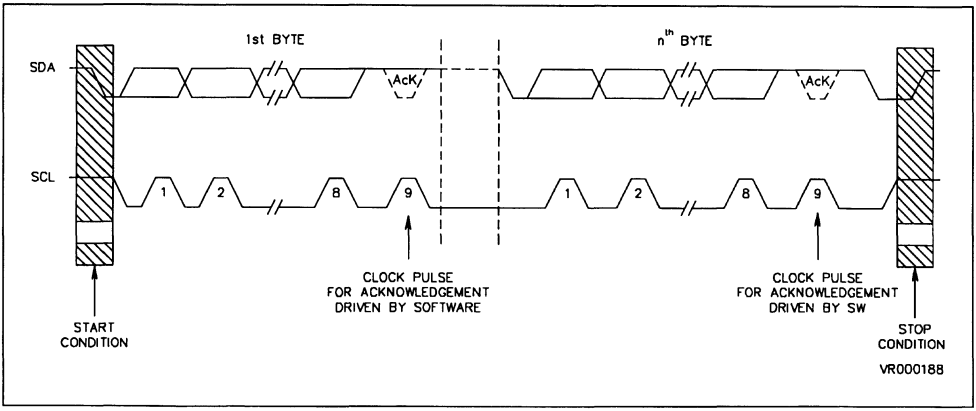


DIFFERENT PROTOCOLS (Continued)

Table 8-2. Typical I^2C-bus Sequences

Phase	Software	Hardware	Notes
INITIALIZE	SPICR.CPOL, CPHA = 0, 0 SPICR.SPEN = 0 SPICR.BMS = 1 SCK pin set as AF output SDI pin set as input Set SDO port bit to 1	SCK, SDO IN HI-Z SCL, SDA = 1, 1	Set polarity and phase SPI disable START/STOP interrupt Enable
START	SDO pin set as output Open Drain Set SDO port bit to 0	SDA = 0, SCL = 1 interrupt request	START condition receiver START detection
TRANSMISSION	SPICR.SPEN = 1 SDO pin as Alternate Function output load data into SPIDR	SCL = 0 Start transmission interrupt request	Managed by interrupt routine load FFh when receiving end of transmission detection
ACKNOWLEDGE	SPICR.SPEN = 0 Poll SDA line Set SDA line SPICR.SPEN = 1	SCK, SDO in HI-Z SCL, SDA = 1 SCL = 0	SPI disable only if transmitting only if receiving only if transmitting
STOP	SDO pin set as output Open Drain SPICR.SPEN = 0 Set SDO port bit to 1	SDA = 1 interrupt request	STOP condition

Figure 8-6. SPI Data and Clock Timing



DIFFERENT PROTOCOLS (Continued)

The data on the SDA line is sampled with the low to high transition on the SCL line.

SPI Working With I²C-bus

To use the SPI with the I²C-bus protocol, the SCK line is used as SCL, the SDI and SDO lines, externally wired-OR'd, are used as SDA. All the output pins must be configured as open drain (see Figure 8-6).

Table 8-2 shows the typical I²C-bus sequence divided in 5 phases: initialize, start, transmission, acknowledge and stop.

Software and hardware will take care of each phase. A master to slave transmission can be managed as example according to the following table.

During the transmission phase, the following I²C-bus features are also supported by hardware.

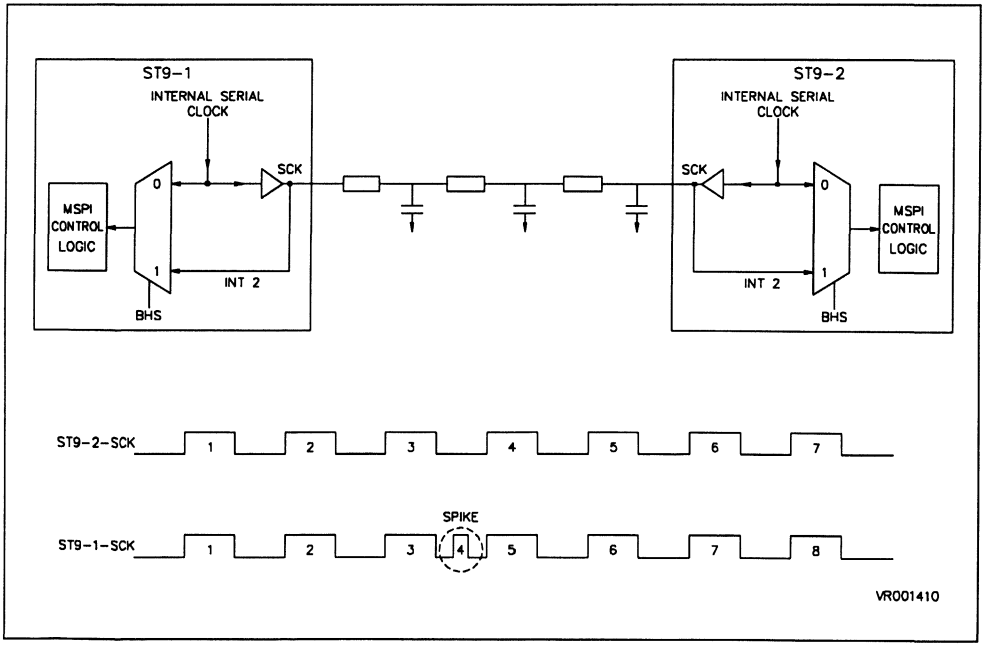
Clock Synchronization

In a multimaster I²C-bus system, when more masters generate their own clock, synchronization is needed. The first master which releases the SCL line stops internal counting, restarting only when the SCL line goes high (released by all the other masters). In this way, devices using different clock sources and different frequencies can be interfaced.

Arbitration Lost

When more masters are sending data on SDA line, the following mechanism is performed: if the transmitter sends a "1" and SDA line is forced low by another device the ARB flag (SPICR.5) is set and the SDO buffer is "switched off. (ARB is reset and SDO buffer is "switched on" when SPIDR is written to again). When BMS is set to "1" the peripheral clock is supplied through the INT2 line by the external clock line (SCL). Due to potential noise spikes (which must last longer than one INTCLK period to be detected), RX or TX may gain a clock pulse.
Referring to Figure 8-7, if ST9-1 detects a noise spike and gains a clock pulse, it will stop its transmission in advance and hold the clock line low causing ST9-2 to be frozen at the 7th bit. To exit and recover from this condition the BMS bit must be reset to "0", this will cause the reset of the SPI logic, aborting the current transmission. An End of Transmission interrupt is generated after this reset sequence.

Figure 8-7. SPI Arbitration



DIFFERENT PROTOCOLS (Continued)

8.5.2 S-Bus Interface

S-bus is a three-wire bidirectional data-bus, with functional features similar to I^2C-bus. Differently from I^2C-bus, the START/STOP conditions are given by encoding the information on 3 wires instead of 2, as shown in Figure 8-8. The additional line is referred as SEN.

SPI Working With S-bus

The S-bus protocol uses the same pin configuration as I^2C-bus for generating the SCL and SDA lines. The additional SEN line is managed through a standard ST9 I/O port under software control (see Figure 8-9).

Figure 8-8. Mixed S-bus and I^2C-bus system

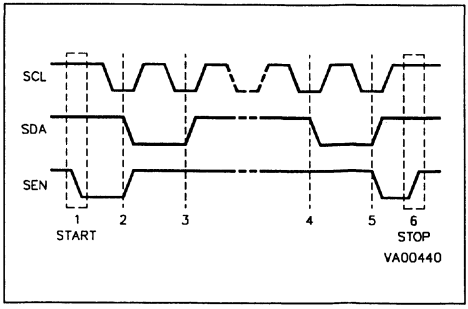


Figure 8-9. S-bus Configuration

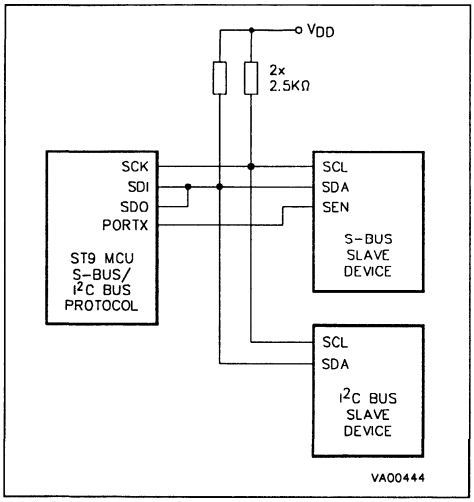
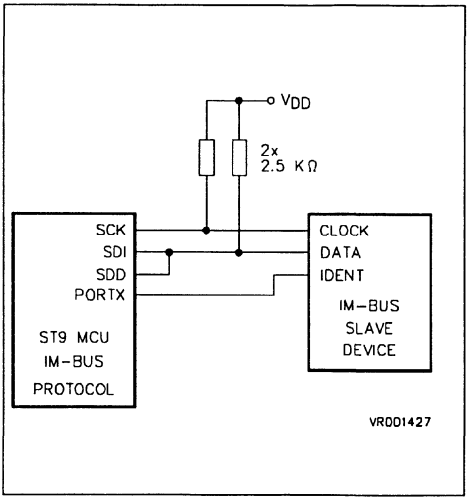


Figure 8-10. ST9 and InterMetal Peripheral



DIFFERENT PROTOCOLS (Continued)

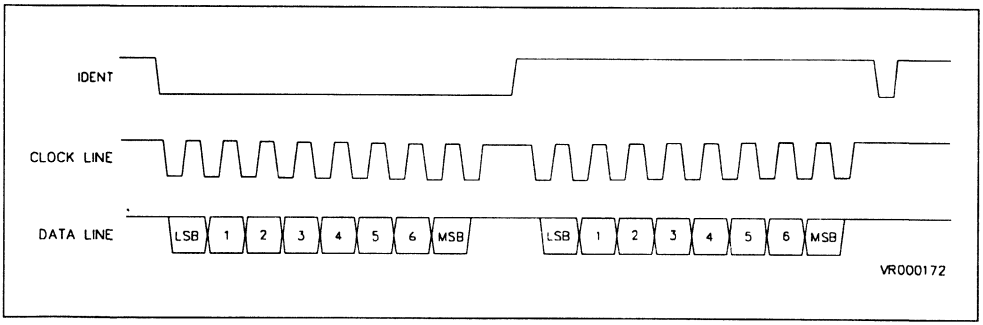
8.5.3 IM-Bus Interface

The IM-bus has a bidirectional data line and a clock line, and in addition it requires an IDENT line that distinguishes an address from a data byte (Figure 8-11). Unlike the I^2C-bus protocol, the IM-bus protocol sends the least significant bit first, this requires a software routine which reverses the bit order before sending, and after receiving a data byte. Figure 8-10 shows the connections for an IM-bus peripheral to an ST9 SPI. The SDO and SDI pins are connected to the bidirectional data pin of the peripheral device. The SDO alternate function is set in Open Drain (external 2.5KΩ pull-up resistors are required).

With this type of configuration, data is sent to the peripheral by writing the data byte to SPIDR. To receive data from the peripheral, the User should write FFh into SPIDR in order to generate the shift clock pulses. As the SDO line is set to the Open Drain configuration, the incoming data bits that are set to one do not affect the SDO/SDI line status (which defaults to a high level due to the FFh in the transmit register), while incoming bits that are set to "0" pull the input line low.

In software it is necessary to initialise the ST9 SPI with CPOL and CPHA set to "1", "1". By using a general purpose I/O as the IDENT line and forcing it to a logical "0" when writing to SPIDR, an address is sent (or read). Then, by setting this bit to a logical "1" and writing to SPIDR, data is sent to the peripheral. When all the address and data pairs are sent it is necessary to drive the IDENT line low and high to create a short pulse. In this way the stop condition is generated.

Figure 8-11. IM bus Timing



NOTES

9 TIMER/WATCHDOG

9.1 INTRODUCTION

A programmable 16-bit down counter with an 8-bit prescaler is included in the ST9294 Core. This Timer can be programmed to be used as a general purpose 16-bit Timer, or as a Watchdog Timer offering security against possible processor malfunctions due to hardware or software failures.

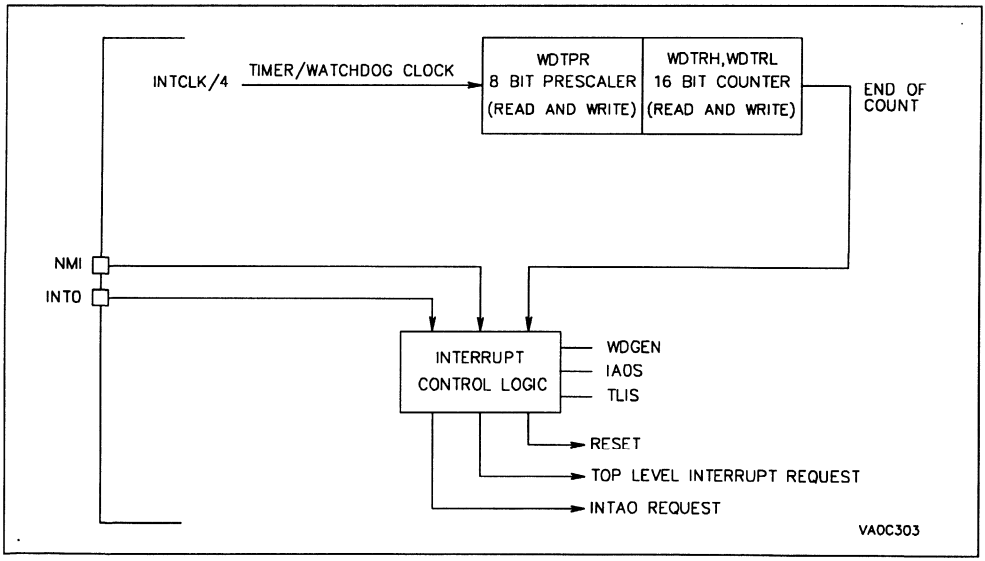
An interrupt generated by the unit (when running as a 16-bit Timer/counter and not as Watchdog) can be used as a Top Level Interrupt or as an interrupt source connected to channel A0 of the external interrupt structure (replacing the INT1 interrupt input).

The clock for the counter is an internal clock equal to INTCLK divided by 4.
When using an external 24MHz crystal (INTCLK = 12MHz), the End Of Count rate is:

5.59 sec. for Max. Count (Timer Const. = FFFFh, Prescaler Const. = FFh)

333 nsec. for Min. Count (Timer Const. = 0000h, Prescaler Const. = 00h)

Figure 9-1. Block Diagram



9.2 FUNCTIONAL DESCRIPTION

9.2.1 Timer/Counter Control

Start/Stop

ST_SP (WDTCR.7) enables down-counting. An instruction which sets this bit will cause the Timer to start at the beginning of the following instruction. Resetting this bit will stop the counter.

If the counter is stopped and restarted, counting will resume from the last value unless a new constant has been entered in the Timer registers. A new constant can be written with the counter running. The new value will be loaded at the following End Of Count (EOC).

WARNING: *In order to prevent incorrect counting of the Timer/Watchdog, the prescaler (WDTPR) and counter (WDTRL, WDTRH) registers must be initialised before the starting of the Timer/Watchdog. If this is not done, counting will start with the reset (un-initialised) values.*

Single/Continuous Mode

SINGLE MODE: At End Of Count the Timer stops, reloads the constant, and resets the Start/Stop bit (WDTCR.6) (user may check the current status by reading this bit). Restarting is done by setting the Start/Stop bit. Note that the Timer constant is reloaded only if it has been modified during the stop period.

CONTINUOUS MODE: At End Of Count the counter automatically reloads the constant and restarts. It is stopped only if the Start/Stop bit is reset. This Mode bit can be written with the Timer stopped or running. It is possible to toggle the S_C bit and start the counter with the same instruction.

9.2.2 Timer/Watchdog Mode

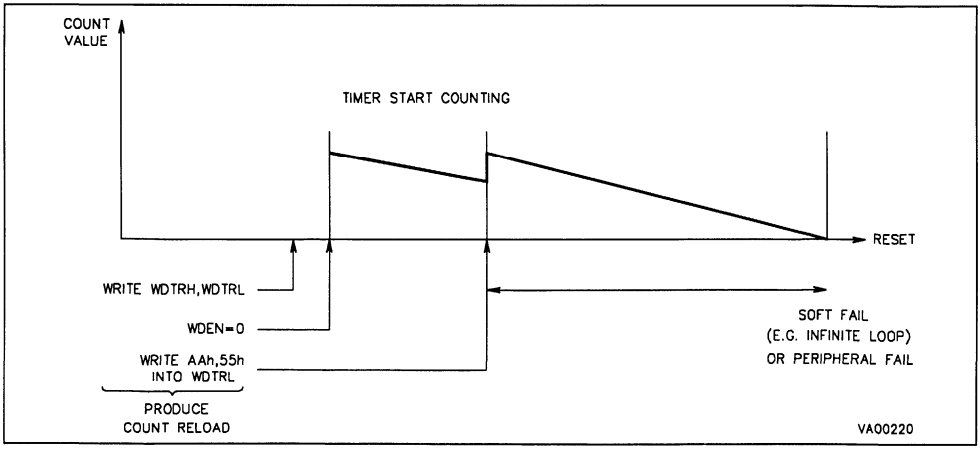
In this mode (WDGEN = "0") the counter generates a fixed time basis. When End Of Count is reached the Timer generates a system Reset.

The time base is user-defined and must be written in the Timer registers before entering Watchdog mode. In Watchdog mode it is possible to modify only the Prescaler Constant. This new value will be loaded when the counter restarts.
Resetting WDGEN (bit 6 of the Wait Control Register) causes the counter to start regardless of the value of the Start-Stop. In order to prevent a system reset the sequence AAh, 55h should be entered in WDTLR (Watchdog Timer register low). Once the writing of 55h has been performed the Timer reloads the constant and counting restarts from the preset value.
The minimum time between the writing of the AAh and 55h codes is zero, i.e. the writing is sequential, and the maximum time is given by the Watchdog timeout period.

In Watchdog-mode a `halt` instruction is regarded as illegal. Execution of the `halt` instruction stops further core execution by the CPU and interrupt acknowledgment, but does not stop INTCLK or CPUCLK or the Watchdog Timer, which will cause a System Reset when reaching the End of Count. Furthermore ST_SP, S_C and input mode selection bits are "don't-care". Hence regardless of their status, the counter always runs in Continuous Mode driven by the internal clock.

The Output mode should not be enabled since that particular mode of operation is meaningless.

Figure 9-2. Timer /Watchdog in Watchdog Mode



FUNCTIONAL DESCRIPTION (Continued)

9.3 TIMER/WATCHDOG INTERRUPT

When enabled, the Timer/Watchdog will issue an interrupt request at every End Of Count.

A pair of control bits, IA0S (EIVR.1, Interrupt A0 selection bit) and TLIS (EIVR.2, Top Level Input Selection bit) allow the selection of 2 interrupt sources (the Timer/Watchdog End of Count or an external pin) in two different ways, as a top level non maskable interrupt (Software Reset) or as a source for channel A0 of the external interrupt logic.

In the Watchdog mode the End Of Count always causes a system reset.

A block diagram of the interrupt logic is given in Figure 9-3 (Note: software traps can be generated by setting the appropriate interrupt pending bit):

The following table shows all the possible configurations of the interrupt/reset sources which involve the Timer/Watchdog:

Figure 9-3. Interrupt Sources

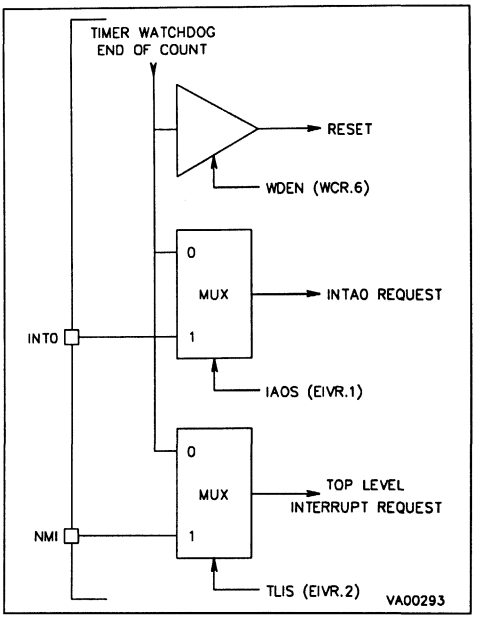


Table 9-1. Interrupt Configuration

Control Bits			Enabled Sources			Watchdog Timer Status
WDGEN	IAOS	TLIS	Reset	INTA0	Top Level	
0	0	0	WDG/Ext Reset	SW TRAP	SW TRAP	Watchdog
0	0	1	WDG/Ext Reset	SW TRAP	Ext Pin	Watchdog
0	1	0	WDG/Ext Reset	Ext Pin	SW TRAP	Watchdog
0	1	1	WDG/Ext Reset	Ext Pin	Ext Pin	Watchdog
1	0	0	Ext Reset	Timer	Timer	Timer
1	0	1	Ext Reset	Timer	Ext Pin	Timer
1	1	0	Ext Reset	Ext Pin	Timer	Timer
1	1	1	Ext Reset	Ext Pin	Ext Pin	Timer

Note.
WDG = Watchdog function
SW TRAP = Software Trap

9.4 TIMER/WATCHDOG REGISTERS

The Timer/Watchdog has 4 registers mapped into Group F, Page 0 of the Register File.

WDTHR (R248): Timer/Watchdog Counter High Register
WDTLR (R249): Timer/Watchdog Counter Low Register
WDTPR (R250): Timer/Watchdog Prescaler Register
WDTCR (R251): Timer/Watchdog Control Register

Three additional control bits are mapped in the following registers of Page 0:

- watchdog mode enable, WCR.6
- top level interrupt selection, EIVR.2
- interrupt A0 channel selection, EIVR.1

Note: The registers containing these bits also contain other functions. Only the bits relevant to the operation of the Timer/Watchdog are shown here.

Counter Registers

This 16 bit register is used to load the 16 bit counter value. The registers can be read or written "on the fly".

WDTHR R248 (F8h) Page 0 Read/Write Timer/Watchdog Counter Register, High byte

Reset value: undefined

7							0
R15	R14	R13	R12	R11	R10	R9	R8

WDTLR R249 (F9h) Page 0 Read/Write Timer/Watchdog Counter Register, Low byte.

Reset value: undefined

7							0
R7	R6	R5	R4	R3	R2	R1	R0

WDTPR R250 (FAh) Page 0 Read/Write Timer/Watchdog Prescaler Register

Reset value: undefined

7							0
PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0

b7-b0 = **PR7-PR0:** *Timer/Watchdog Prescaler.* The value stored in this Register is used to select the prescaling factor from 1 (loading 00h) to 256 (loading FFh).

WARNING. *In order to prevent incorrect counting of the Timer/Watchdog, the prescaler (WDTPR) and counter (WDTRL, WDTRH) registers must be initialised before the starting of the Timer/Watchdog. If this is not done, counting will start with the reset (un-initialised) values.*

WDTCR R251 (FBh) Page 0 Read/Write Timer/Watchdog Control Register

Reset value: 0001 0010 (12h)

7							0
ST_SP	S_C	INMD1	INMD2	INMD3	INMD4	INMD5	INMD6

b7 = **ST_SP:** *Start/Stop Bit.* Setting this bit to a "1" starts the counting operation (see Warning above). When this bit is "0", the counter is stopped (reset status)

b6 = **S_C:** *Single/Continuous.* When this bit is set, the counter operates in Single Count Mode. Continuous Mode is set when this bit is "0"

b5-b0 = **INMD1, INMD6:** Reserved bits must be held in their reset status

TIMER/WATCHDOG REGISTERS (Continued)

WCR R252 (FCh) Page 0 Read/Write
Wait Control Register

Reset value: 0111 1111 (7Fh)

7							0
X	WDGEN	X	X	X	X	X	X

b6 = **WDGEN:** *Watchdog Enable Bit (active low).* Resetting this bit to zero via software enters the Watchdog mode. Once reset, it cannot be set to "1" by the user program. At system reset, the Watchdog mode is disabled

EIVR R246 (F6h) Page 0 Read/Write
External Interrupt Vector Register

Reset value: xxxx 0110 (X6h)

7							0
X	X	X	X	X	TLIS	IAOS	X

b2 = **TLIS:** *Top Level Input Selection bit.* This bit selects the Top Level interrupt source. When "0", the Top Level interrupt source is the Watchdog/Timer end of count, when = "1", it is the external pin NMI.

b1 = **IAOS:** *Interrupt channel A0 Selection Bit.* This bit allows the Timer/Watchdog interrupt to channel through the external Interrupt A0 source, allowing the setting of user-defined priority levels.

WARNING. *To avoid spurious interrupt requests, an access to the IAOS bit must be made only when the interrupt logic is disabled (i.e. after the DI instruction). It is also necessary to clear a possible interrupt pending request on channel A0 before enabling this interrupt channel. A delay instruction (e.g. a NOP instruction) must be inserted between the reset of the interrupt pending bit and the IAOS write instruction.*

NOTES

10 SLICE TIMER

10.1 INTRODUCTION

The Slice Timer unit is a programmable 16-bit down counter and an associated 8-bit prescaler with Single and Continuous counting modes capability. It is similar in function to the Timer/Watchdog.

This document relates to ST9 devices which do not have the external Slice Timer input and output pins available. This implies that certain control bits have to be programmed with a mandatory value and that others are don't care.

10.2 SLICE TIMER-COUNTER FUNCTIONS

The Slice Timer clock is provided by an internal clock equal to INTCLK divided by 4. Thus when a 24MHz crystal is used, a 3MHz maximum counting frequency can be reached or 2MHz with a 8MHz crystal and the divider by 2 disabled. The minimum and maximum counting period are:

- 5.59s for Maximum Count (Timer Const = FFFFh, Prescaler Const = FFh)

- 333ns for Minimum Count (Timer Const = 0000h, Prescaler Const = 00h)

End of Count condition is defined as the Counter Overflow, whenever 00h is reached.

10.2.1 Working Modes

Start-stop Count. The ST-SP bit (STCR.7) is used in order to start and stop counting. An instruction which sets this bit will cause the Slice Timer to start at the beginning of the next instruction. Resetting this bit will stop the counter.

If the counter is stopped and restarted, counting will resume from the value held at the stop condition, unless a new constant has been entered in the Slice Timer registers during the stop period. In this case, the new constant will be loaded as soon as counting is restarted.

WARNING: *In order to prevent incorrect counting of the Slice Timer, the prescaler (STPR) and counter (STLR, STHR) registers must be initialised before the starting of the timer. If this is not done, counting will start with the reset (un-initialised) values.*

Single/continuous Mode. The S-C bit (STCR.6) selects between the Single or Continuous mode.

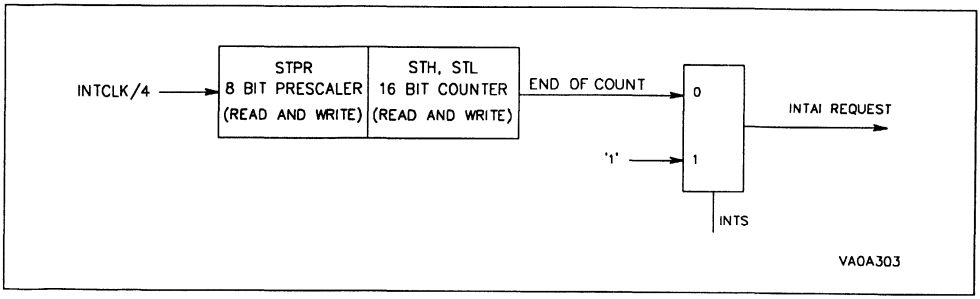
SINGLE MODE: at the End of Count, the Slice Timer stops, reloads the constant and resets the Start/Stop bit (the user programmer can inspect the timer current status by reading this bit). Setting the Start/Stop bit will restart the counter.

CONTINUOUS MODE: At the End of the Count, the counter automatically reloads the constant and restarts. It is only stopped by resetting the Start/Stop bit.

The S-C bit can be written either with the timer stopped or running. It is possible to toggle the S-C bit and start the Slice Timer with the same instruction.

Input Modes. Bits INMD2, INMD1 and INEN are used to select the input modes. Input Enable (INEN) bit enables the input mode selected by the INMD2, INMD1 bits. If the input is disabled (INEN = "0"), the values of INMD2 and INMD1 are not taken into account. In this case, this unit acts as a 16-bit timer (plus prescaler) directly driven by INTCLK/4.

Figure 10-1. Slice Timer Block Diagram



SLICE TIMER (Continued)

10.2.2 Interrupt Selection

The Slice Timer may generate an interrupt request at every End of Count. The STCR bit 2 (INTS) selects the interrupt source between the Slice Timer interrupt and an external interrupt (connected to Vcc for the ST9294). The Slice Timer Interrupt uses the INT1 interrupt channel and takes the priority and vector of the INTA1 external interrupt channel.

If INTS is set to "1", the Slice Timer interrupt is disabled; otherwise, an interrupt request is generated at every End of Count.

Care must be taken when disabling the Slice Timer Interrupt as a rising edge may be generated on the INTA1 channel, causing an unwanted interrupt.

10.2.3 Slice Timer Registers

This unit has 4 registers mapped into the page 0Bh in Group F of the Register File:

Register Address	Register	Function
F0	STH	Counter High-Byte Register
F1	STL	Counter Low-Byte Register
F2	STP	Slice Timer Prescaler Register
F3	STC	Slice Timer Control Register
F4-FF	—	Reserved

10.2.4 Register Description

STH R240 (F0h) Page 0B Read/Write
Counter High-Byte Register

Reset value: undefined

7							0
ST.15	ST.14	ST.13	ST.12	ST.11	ST.10	ST.9	ST.8

b7-b0 = **ST.15-ST.8:** *Counter High-Byte.*

STL R241 (F1h) Page 0B Read/Write
Counter Low-Byte Register

Reset value: undefined

7							0
ST.7	ST.6	ST.5	ST.4	ST.3	ST.2	ST.1	ST.0

b7-b0 = **ST.7-ST.0:** *Counter Low-Byte.* Writing to the STH and STL registers allows the user to enter the Slice Timer constant, while reading provides the counter current value. Thus it is possible to read the counter on-the-fly.

STP R242 (F2h) Page 0B Read/Write
Slice Timer Prescaler Register

Reset value: undefined

7							0
STP.7	STP.6	STP.5	STP.4	STP.3	STP.2	STP.1	STP.0

b7-b0 = **STP.7-STP.0:** *Prescaler.* The Prescaler value for the Slice Timer is programmed into this register. When reading the STPR register, the returned value corresponds to the programmed data instead of the current data.

STC R243 (F3h) Page 0B Read/Write
Slice Timer Control Register

Reset value: 000x x1xx

7							0
ST-SP	S-C	INMD1	INMD2	INEN	INTS	OUTMD1	OUTMD2

b7 = **ST-SP:** *Start-Stop Bit.* Setting ST-SP to "1" starts the counting operation. Writing "0" stops the Slice Timer.

b6 = **S-C:** *Single-Continuous Mode Select.* Setting S-C to "1" sets the Slice Timer to Single Mode. Writing "0" sets the Continuous Mode (Reset Status)

b5-b4 = **INMD1, INMD2:** *Input Mode Selection.* These bits are don't care as external pins are not available.

b3 = **INEN:** *Input Enable.* INEN must be set to "0" (Input section disabled).

b2 = **INTS:** *Interrupt Selection.* Setting INTS to "1" disables the Slice Timer interrupt (the Reset status).
Writing "0" enables the Slice Timer interrupt on channel INT1.

b1-b0 = **OUTMD1, OUTMD2:** *Output Mode Selection.* These bits are don't care.

11 ON SCREEN DISPLAY

11.1 INTRODUCTION

The ST9 On-Screen Display (OSD) is used to display lines of characters under control of the ST9 CPU on a Video Display screen (for example close caption displays on a television). The screen insertion of the displayed characters is fully synchronised by the vertical and horizontal TV synchronization signals. The OSD generates the Red, Green, Blue and Fast blanking video signals.

The OSD displayable characters are located into the character ROM memory. The set of 128 different characters is fully user definable by ROM mask option.

To give more flexibility for using the OSD, a buffered one-row RAM architecture is used.

Only one row of characters is described at a time within the one-row RAM memory. This one-row Display RAM contains the description of the currently displayed row: row and word attributes, horizontal placement and character codes which will be converted into pixel information issued from the character ROM memory.

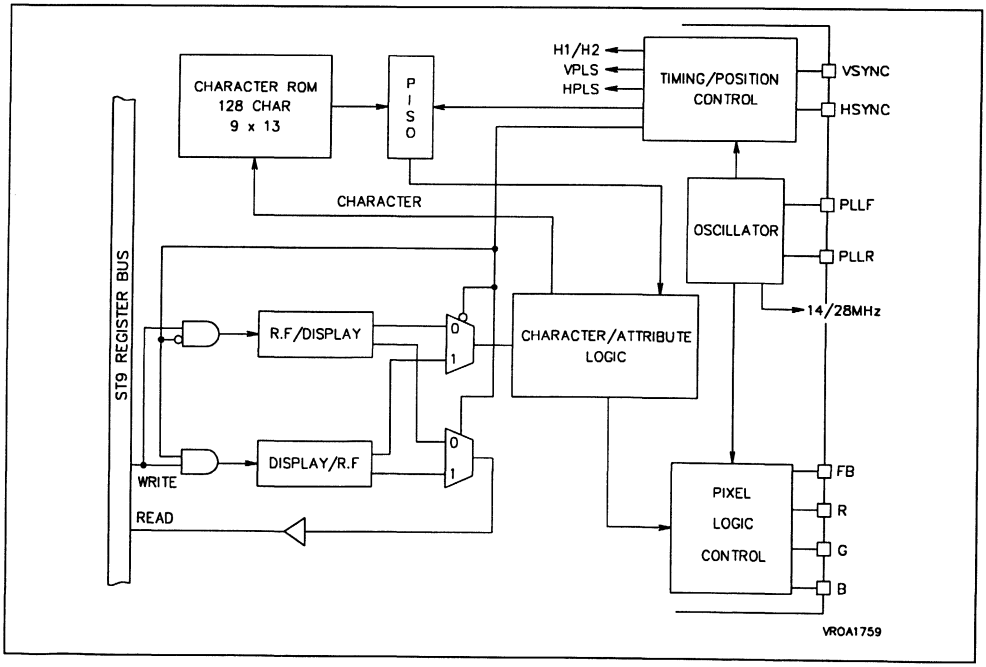
During the time the Display RAM is displaying on the screen, the CPU can prepare the content of the next row of characters by re-writing the OSD RAM registers. At the time the next row must be displayed the RAM register content is swapped to the one-row Display RAM, allowing the CPU to prepare the further row content and location.

An interrupt request may be generated each time the swap occurs (i.e. each time the one-row RAM content swapped to the display function).

The interrupt request uses the INT4 channel, taking the priority and mask of INTC0 and the vector programmed in EIVR.

Other information such as the vertical placement of the next row of characters is given as a vertical ad-

Figure 11-1. OSD Block Diagram



ON SCREEN DISPLAY (Continued)

dress by programming the Event Line register. The real-time vertical position is calculated by the Scan Line counter which counts the TV horizontal synchronisation pulses. Then, at the time the Scan Line counter matches the Event Line value, the swap mechanism is activated.

This architecture allows the CPU to perform vertical scrolling of the rows by software.

11.1.1 Format

Each row is able to display 34 characters in small size mode.

The row RAM size is 36 bytes. The two first locations are not displayed but are used as attribute locations.

Each character is composed by a 9x13 pixel matrix.

Each pixel is at least 2 clock periods wide and 2 lines high. The pixel clock is generated by an internal controlled oscillator, whose frequency is locked to the horizontal synchronisation signal as follows:

fosc = 896 x F(HSYNC)

The pixel pattern of the character is stored in the character ROM.

The colour controller displays the foreground colour when a "1" is read from the ROM or displays the background colour when a "0" is read.

These attributes can be modified by the rounding and fringe logic.

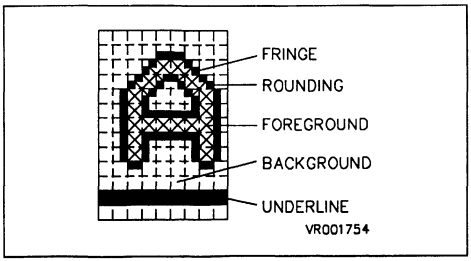
As shown in Figure 11-2 the rounding mechanism is done by adding a half-dot to smooth the diagonal lines. The half-dot is painted as foreground.

The fringe is a half-dot wide black border. Fringe and rounding may be turned on by setting word attribute bits.

All the screen area outside of the displayed rows of characters is displayed using the border colour.

The colour of background, foreground and border are sent to the OSD controller as attributes.

Figure 11-2. Character Formats



11.1.2 Attributes

The attributes of the OSD are serial: Two kinds of attributes are available:

- Row attributes which apply to the entire row.
- Serial attributes which apply until changed, or to the end of the line.

Row attributes are defined by two registers. These 2 registers are copied on each swap (when the RAM registers are switched to the one-row Display RAM). The modifications made in the row attribute registers are active during the next row and the following if no new write occurs.

The row attribute registers are initialized by the CPU reset condition.

Row attributes 1 register:

HDCS R244 (F4h) Page 2A Read/Write
Horizontal Delay/Character Size Register

Reset value: 0000 0000b (00h)

7							0
SH	SW	HD5	HD4	HD3	HD2	HD1	HD0

The HDCS Register sets the character size for the display row, and the horizontal delay for positioning the start of the row on the screen relative to the start of the raster line.

b7-b6 = **SH, SW:** *Pixel Size.* These two bits are encoded to select the pixel size for the displayed characters.

Pixel Size for Displayed Characters (Single Scan)

SH	SW	Pixel Height Horizontal lines/field	Pixel Width		Max. Char. /Row
			#Clock	H	
0	0	1	2	1H	34
0	1	2	2	2H	34
1	0	2	3	1.5H	24
1	1	2	4	2H	18

b5-b0 = **HD5-HD0:** *Horizontal delay.* The starting position on the video line can be offset by programming the HD value. This value is loaded into a down-counter using the OSD clock as input at the beginning of the line. When the counter reaches 0, the display will start (beginning with the two undisplayed characters).

ON SCREEN DISPLAY (Continued)

The total delay from the active edge of HSYNC to the position of the first displayed character is equal to:

86 x T_{osc} + [HDel x (HDRES+1)+18] x T_{pixel}

Where T_{pixel} (width of one pixel) is:

$2T_{osc}$ for SH, SW= 0, X

$3T_{osc}$ for SH, SW= 1, 0

$4T_{osc}$ for SH, SW= 1, 1

HD_{RES} = "0" for small resolution (refer to the Enable Register description)

Row attributes 2 register:

AR R245 (F5h) Page 2A Read/Write
Active Range Register

Reset value: 0000 0000b (00h)

7							0
RS3	RS2	RS1	RS0	RE3	RE2	RE1	RE0

b7-b4 = **RS3-RS0:** *Active Range Start Value.*

b3-b0 = **RE3-RE0:** *Active Range Stop Value.*

These two values are compared to the vertical pixel character counter value. The RS range is 0 to 12 and RE range is 1 to 13.

The counter is reset to 0 at the beginning of each row and counts each vertical pixel row up to 12.

If the value of the counter is outside the range (less than RS3-RS0 or greater than RE3-RE0) the OSD displays the border attributes, otherwise the display is normal.

For example:

If RS = 0 and RE = 1, only the first line of the row is displayed.

This feature is useful for the software generated vertical smooth scrolling, a row of characters can appear or disappear line by line.

11.1.3 Serial Attributes:

The serial attributes registers describe how the current row is built. There are stored within the displayed line buffer in page 28h: F0h-FFh, page 29h: F0h-FFh and page 2Ah: F0h-F3h (refer to the OSD Memory Map).

The serial attribute is sent to the OSD inside the display RAM. When a valid serial attribute code is encountered the corresponding serial attribute register is loaded with the new value, and remains in effect until the end of the row, or the next serial attribute.

If the most significant bit of the RAM location is "0", the seven last significant bits have the meaning of the code of the character, if it is "1", the byte contains a serial attribute.

In this case the OSD displays a space and stores the new attributes. These attributes will remain active until a new value is encountered.

The first two locations of the row RAM have a special behaviour. They are displayed as border, but if they contain a valid serial attribute (MSB = "1") the corresponding attribute register is modified.

Two sets of attributes can be stored using bit 6 as address bit:

If b6 = "0" the "foreground attributes" are modified.

If b6 = "1" the "additional attributes" are modified.

Character code byte:
(b7 = "0")

7							0
0	C6	C5	C4	C3	C2	C1	C0

Character 00h displays the border colour. Normally the border colour begins at the first column of the character matrix and is a full 9 pixels width. However, there is an exception to this rule. If the character immediately preceding the 00h is in the range of 01h-7Fh, and the italics word attribute is on, the previous character will extend half way into the 00h. In this case, the border colour will begin in the middle of the character matrix and extend to the right edge of the matrix. The right edge of character 00h is never extended into the next character.

Characters 01h-7Fh are the printable characters. Normally they will display the pixel pattern defined in the character ROM, and will be 9 pixels wide. However, there is an exception to this rule. If the italics serial attribute is on, the character will be slanted to the right.

Each character in a displayed word will be 9 pixels wide except for the last. This will have the matrix extended to the right, half way into the next character. This is true regardless of what the next character is, unless it is the last character to be displayed. In this case, it will be truncated to 9 pixels wide.

The serial attributes: flash, underline, round, and fringe also effect these characters.

ON SCREEN DISPLAY (Continued)

Foreground attributes byte:
(b7 = "1", b6 = "0")

7							0
1	0	FLA	ITA	UND	FOR	FOB	FOG

b6 = **FLA:** *Flash Enable.* FLA = "1" forces the flashing mode for the following word. According to the value of the "flash on" bit (located into the Enable register), the word is displayed as in normal mode (FON = "0") or displayed as a space with background colour (FON = "1").

Underline also flashes.

b5 = **ITA** *Italic Enable.* ITA = "1" forces the following characters to be displayed with the Italic attribute.

Italic is displayed by applying a slant to the character. The slope is equal to 2.5 pixels horizontally for the 13 vertical pixels. The horizontal shift for each line is the following:

lines 0-1 : 2.5p lines 6-7 : 1p
lines 2-3 : 2p lines 8-9 : 0.5p
lines 4-5 : 1.5p lines 10-12 : 0p

The mandatory spaces needed to set and reset the italic attribute are used to solve the border problems between italic and the normal character shape.

b3 = **UND:** *Underline enable.* UND = "1" forces the underlining of the following word. The 11th (relative to 0) row of character pixels is painted as foreground. The Underline affects only words (spaces are not underlined).

b2-b0 = **FOR, FOG, FOB:** *Foreground colour select.* FOR, FOG and FOB bit control the RGB output for the foreground colour.

FOR	FOG	FOB	Foreground colour
0	0	0	Black
0	0	1	Blue
0	1	0	Green
1	0	0	Red
0	1	1	Cyan
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

Additional attributes byte:
(b7 = "1", b6 = "1")

7							0
1	1	ROU	FR	BGS	BGR	BGB	BGG

b5 = **ROU:** *Rounding enable.* ROU = "1" enables Rounding of the following characters, ROU = "0" disables character Rounding.

b4 = **FR:** *Fringe enable.* FR = "1" enables the Fringe effect for the following characters, FR = "0" disables the Fringe.

b3 = **BGS:** *Transparency select.* BGS = "1" sets the background to a solid colour defined by BGR, BGB, BGG. If BGS = "0", the background is transparent.

The transparency of the background is modified in the middle of the space containing the serial attribute.

b2-b1 = **BGR, BGB, BGG:** *Background colour select.* BGR, BGB and BGG bit control the RGB output for the background colour. The background colour modification occurs in the middle of the serial attribute space.

The colours generated follow the same coding as shown for FOR, FOG and FOB.

Characters 80h-FFh are control characters, but will be displayed as background colour only. They always display the first half of the character matrix with the previous background colour, and then the second half with the current background colour. If they are immediately preceded by an italic printing character, the italic character will extend half-way into the control character. The right edge of control characters is never extended into the next character. The serial attributes: flash, underline, rounding, and fringe have no effect on control characters.

ON SCREEN DISPLAY(Continued)

Figure 11-3. Italic Attributes

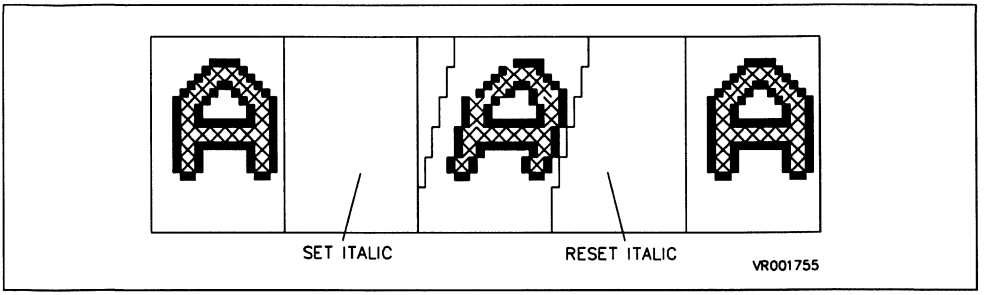
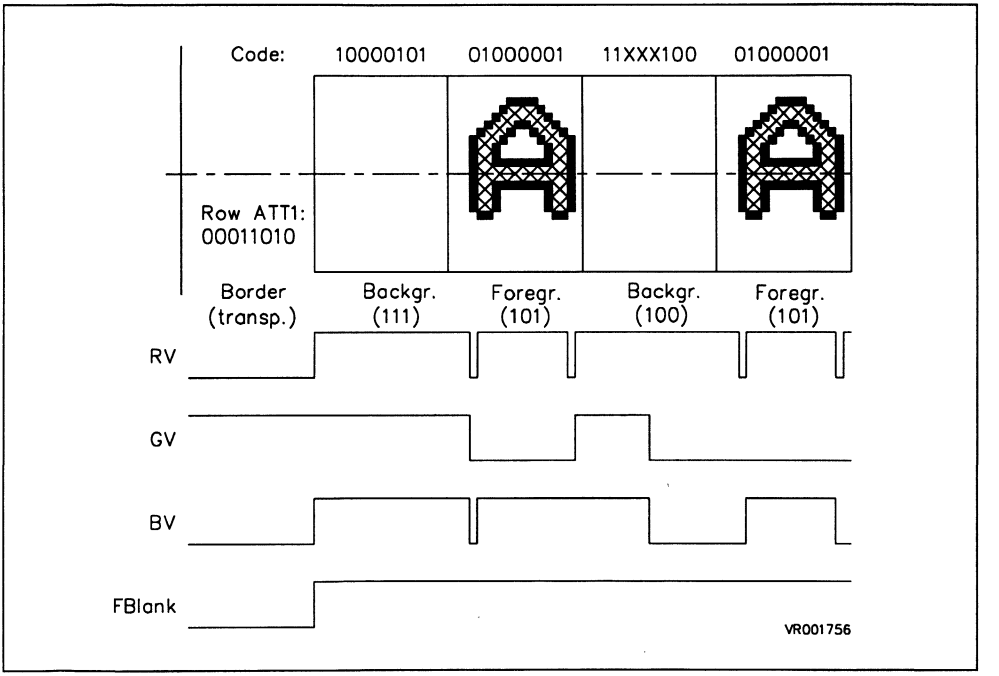


Figure 11-4. Colour Outputs



ON SCREEN DISPLAY(Continued)

11.1.4 Status Registers

The status registers give the programmer a real time status of OSD functions. This data is useful to synchronize the software with the display.

The status registers are READ-ONLY registers.

SL R249 (F9h) Page 2A Read-only
Scan Line Register

Reset value: 0000 000xb (0xh)

7							0
SL7	SL6	SL5	SL4	SL3	SL2	SL1	SL0

b7-b0 = SL7-SL0: *Scan Line Counter.* The nine bit counter SL8-SL0 indicates the vertical position of the beam. This counter starts at 0 at the top of the screen and is incremented after HSY in case of normal scan mode, and after 2 HSY in case of double scan mode (2H).

FB R250 (FAh) Page 2A Read-only
Flag Bit Register

Reset value: 0000 0000b (00h)

7							0
SL8	VSY	HSY	0	VP3	VP2	VP1	VP0

b7 = **SL8:** *Scan Line bit 8.* SL8 is the MSB (ninth) bit of the Scan Line counter (see above).

b6 = **VSY:** *Vertical Sync active.* This bit is a flag activated (set to "1") during the vertical fly-back, information being issued from the VSYNC pad controlled by the programming of the Delay register and the VSYNC polarity (refer to the Border Color register).

b5 = **HSY:** *Horizontal Sync active.* This bit is a flag activated (set to "1") on the starting edge of the HSYNC pad signal. HSY remains active during 4µs in single scan (2µs in double scan mode).

b3-b0 = **VP3-VP0:** *Vertical Sync Phase.* This nibble gives the phase value between the starting edge of the internal HSYNC signal and the starting edge of the first internal VSYNC signal that follows HSYNC. Phase(VSYNC-HYNC) =

$$\frac{\text{VSYNC} - \text{HSYNC}}{4\mu s} + V_D + 1 \leq V_P \leq \frac{\text{VSYNC} - \text{HSYNC}}{4\mu s} + V_D + 2$$

Note: VP3 can be used as a field polarity flag (refer to the Delay register).

11.1.5 Control Registers

The control registers are used by the programmer to control the behavior of the OSD.

BCOL R251 (FBh) Page 2A Read/Write
Border Color Register

Reset value: 0000 0000b (00h)

7							0
HPOL	VPOL F	BPOL	RGBPOL	BOS	BOR	BOB	BOG

b7 = **HPOL:** *HSYNC pad polarity.* HPOL = "0" for positive polarity (i.e. HSYNC = "1" during the line synch pulse).

b6 = **VPOL:** *VSYNC pad polarity.* VPOL = "0" for positive polarity (i.e. VSYNC = "1" during the frame synch pulse).

b5 = **FBPOL:** *Fast blanking polarity.* FBPOL = "0" for positive polarity.

b4 = **RGBPOL:** *RGB outputs polarity.* RGBPOL = "0" for positive polarities.

b3 = **BOS:** *Border Transparency Select.* When BOS = "1" the border is solid (displayed with the colour set by BOR, BOB, BOG).

When BOS = "0" the border is transparent.

b2-b0 = **BOR, BOB, BOG:** *Border colour.* Bor, Bob, Bog bits program the value of the respectively R/B/G outputs when displaying the border.

The value stored in this register is not affected by the row refresh mechanism. There is no need to confirm the value for each new transfer. This can produce a visible parasitic effect on the display.

It is possible to change the border color on the fly during the display.

ON SCREEN DISPLAY(Continued)

EN R247 (F7h) Page 2A Read/Write
Enable Register

Reset value: 0000 000xb (0xh)

7							0
0	TE	FON	OSDE	PLLE	1H/2H	HDRES	EL8

b7 = Read only to "0"

b6 = **TE:** *Transfer Enable.* It is used to enable (TE = "1") or disable (TE = "0") the swap of the RAM registers content to the display RAM each time the Scan Line counter content matches the Event Line register value.

An interrupt request pulse is generated and forwarded to the core each time the match occurs independently to the value of TE.

If TE = "0" no transfer occurs (no swap) and the screen will display border colour.

b5 = **FON:** *Flash On.* FON= "1" Flash on: The characters with the flash attribute are displayed as space. FON = "0" The flash attribute has no effect on the characters. This bit is used to control the flashing period by software.

b4 = **OSDE:** *OSD Enable.* When set it will start the OSD function and allow displaying. When reset, it will stop the OSD clock and force the Blanking ouput in its inactive state; but OSD basic functions such as: Event Line counter, interrupt request generation, Flag Bit and Delay registers are remaining active.

b3 = **PLLE:** *PLL Oscillator Enable.* If set to "1", the internal pixel oscillator is started and runs, if set to "0", the internal oscillator is stopped.

b2 = **1H/2H:** *Scan Select.* "0" for single scan, "1" for double scan (31kHz).

b1 = **HDRES:** *Horizontal Delay Resolution.* If HDRES = "0" "1 pixel" resolution for the Horizontal delay if HDRES = "1" "2 pixel" resolution.

b0 = **EL8:** *Event Line Register bit 8.* EL8 is the MSB (ninth) bit of the Event Line register (see Register EL).

Remark: The control of the video outputs (Fast blanking and RGB) is made using the standard Alternate Function I/O port features (see I/O ports chapter)

EL R246 (F6h) Page 2A Read/Write
Event Line Register

Reset value: 0000 0000b (00h)

7							0
EL7	EL6	EL5	EL4	EL3	EL2	EL1	EL0

b7-b0 = **EL7-EL0:** *Event Line Register.* The Event Line register is written by the CPU to indicate the next vertical location for displaying the row of characters. The accuracy of the position is one line in single scan mode. When the content of the scan line register matches with the content of the event line register an interrupt request pulse is generated and forwarded to the core on the INT4(INTC0) channel.

VD R248 (F8h) Page 2A Read/Write
Vertical Delay Register

Reset value: 1111 0000b (F0h)

7							0
1	1	1	1	VD3	VD2	VD1	VD0

b7-b4 = Read only to "1"

b3-b0 = **VD3-VD0** *Vertical Delay.* This programmable delay is applied on the Vertical Synchronisation input to balance the delay introduced by the scanning processor.

4 bits value programmable delay:

$(V_D \cdot 4\mu s) + 4\mu s \leq d \leq (V_D \cdot 4\mu s) + 8\mu s$ (up 68µs)

Using a quarter of line delay, the distinction between the two fields is possible:

If the active edge of the delayed vertical signal occurs in the first half of the line the parity flag is reset (even field); if it happens during the second half, the flag is set (odd field).

Adding an half line value to the delay of VSYNC leads to an inversion of the odd and even field of the display.

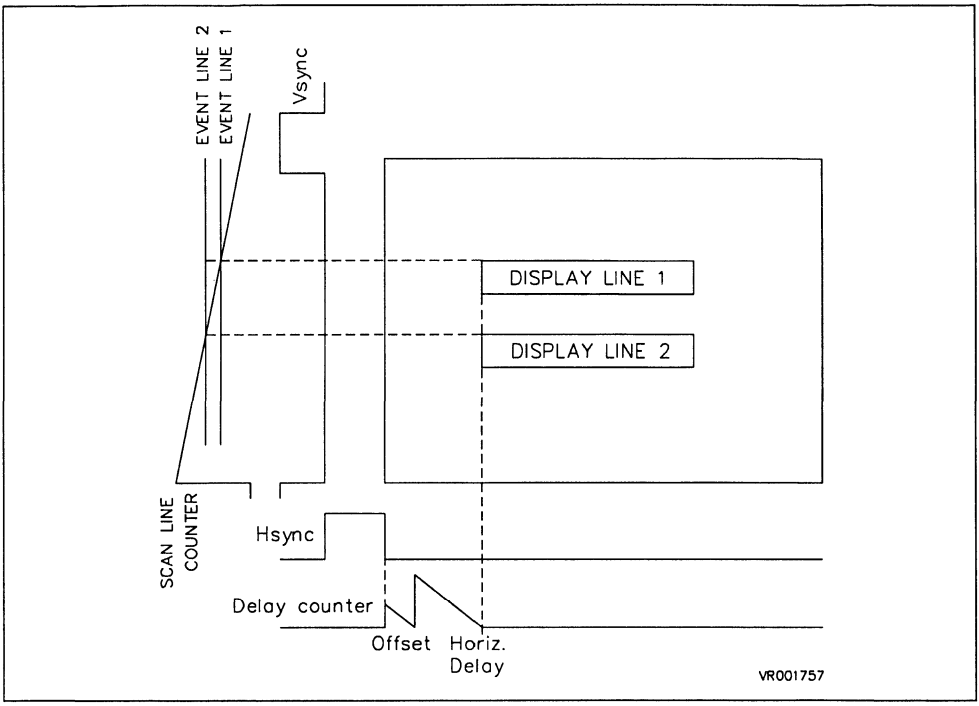
Warning: *Programming the Vertical Delay Register to FFh freezes the scan line counter, disabling any further RGB output. It is Mandatory to always initialise the Vertical Delay Register and to avoid programming it to FFh.*

The value stored in this register is not affected by the row refresh mechanism. There is no need to confirm the value for each new transfer.

This value is used to compensate the delay between HSYNC and VSYNC and must not be updated each frame.

ON SCREEN DISPLAY(Continued)

Figure 11-5. OSD Line Positioning



OSD Input Timing Requirements

Symbol	Parameter	Value		Unit
		Min.	Max.	
f_{OSC}	Oscillator frequency			MHz
T_{VWL}	Vsync pulse width	4		µs
T_{HWL}	Hsync pulse width	4		µs

ON SCREEN DISPLAY(Continued)

11.1.6 RGB Blanking:

When the fast blanking FB is inactive, the RGB outputs are forced to a black level.

If FB and RGB have a positive polarity:
FB = "0" => RGB = "0"

11.1.7 OSD Memory Map

The OSD uses three pages of 16 paged registers.

Register Address.	Page 28
F0	Char. Column 1 displayed as border
F1	Char. Column 2 displayed as border
F2	Char. Column 3 First displayed character
.	.
.	.
FF	Char. Column 16
	Page 29
F0	Char. Column 17
F1	Char. Column 18
.	.
.	.
FF	Char. Column 32
	Page 2A
F0	Char. Column 33
F1	Char. Column 34
F2	Char. Column 35 33th displayed character
F3	Char. Column 36 34th displayed character
F4	Horizontal delay
F5	Active range
F6	Event line
F7	Enable register
F8	Delay register
F9	Scan line
FA	Flag bits
FB	Border colour
FC-FF	Reserved

11.1.8 OSD Interface

OSD interfaces internally with the CPU and externally with the video logic. Furthermore the OSD needs a proper clock fitting with the high frequency video signals. This is internally generated by a fully internal oscillator controlled by a PLL and a few external components as described below.

The video interface is composed by 4 output and 2 input lines. The polarities of these signals are programmed by software.

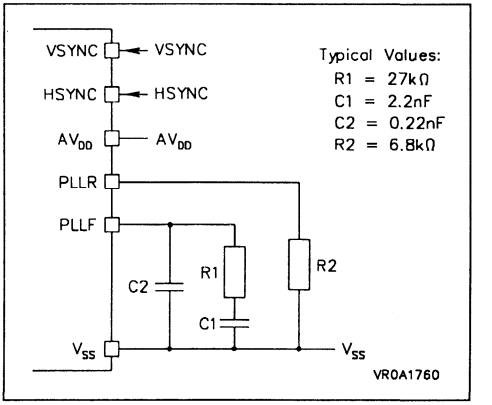
- R, G and B outputs: Color signal outputs. Positive polarity means R=G=B = "1" is white.
- FB output: Fast blanking. Positive polarity means FB = "1" if the video signal is given by the OSD, "0" if transparent (also means FB inactive) When FB is inactive the RGB signals are forced to a black level.

The I/O port bits associated to R,G,B and FB must be set to Alternate Function Output mode for these signals to appear externally.

- VSYNC input: Vertical synch pulse. Positive polarity means VSYNC = "1" during the flyback pulse.
- HSYNC input: Horizontal synch pulse. Positive polarity means HSYNC = "0" during the line (HSYNC = "1" during Horizontal retrace).

Clock pins. Three pins are dedicated to the OSD clock: PLLF, PLLR and AVDD. The clock is synchronized with the HSYNC signal with a PLL; an external loop filter must be connected to the phase comparator output PLLF. A resistor must be connected to PLLR.

Figure 11-6. OSD Oscillator



NOTES

12 DATA SLICER

12.1 INTRODUCTION

12.1.1 Data Slicer Description

The Data Slicer peripheral of the ST9294 is able to extract Closed Caption data from a composite video signal. When used in conjunction with the On Screen Display, the Data Slicer allows for Closed Caption or subtitling to be easily programmed.

The Data Slicer accepts the incoming composite video signal as an AC coupled signal through a 1µF capacitor to the CC Video pin. The OSD synchronisation signals and PLL clock are used in the data extraction.

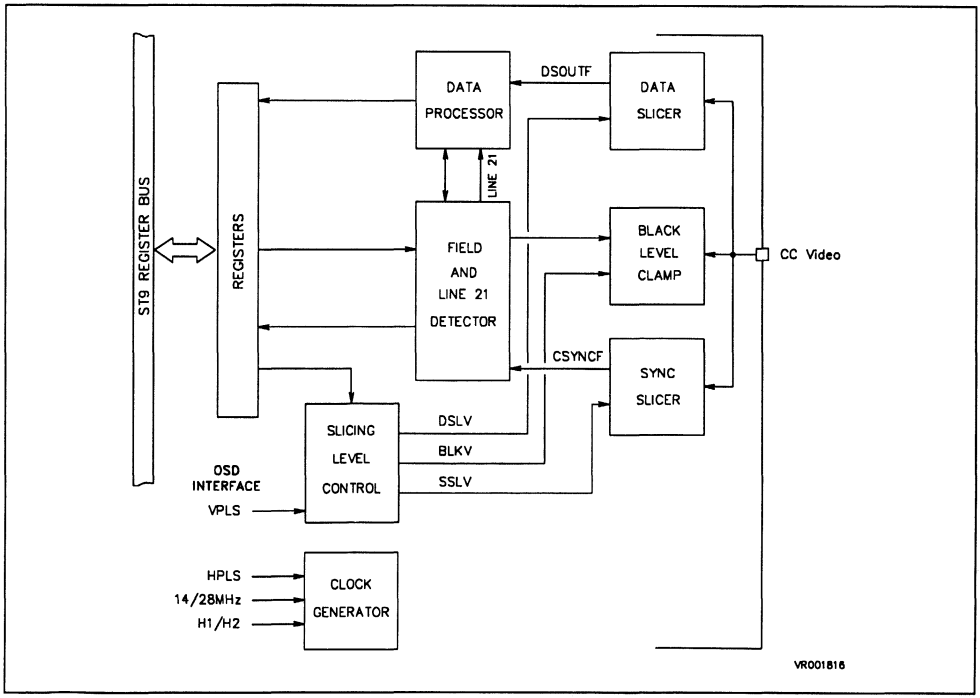
Data Extraction can be programmed for a selectable line in either field, from a video signal of amplitude 1V± 6dB or 2V± 3dB according to the RANGE (R243.3) bit. The slicing levels for data and sync may be software controlled and/or have automatic setting. Status flags are provided to allow the maximum flexibility in operation.

12.1.2 Data Slicer Operation

Data Processing. To enable the Data Slicer the user must set the EDS flag (R244.7); when EDS = "0" the Data Slicer clock is stopped and some flags are forced in their reset state (see register description).

Four clock frequencies are generated by the Clock Generator starting from the OSD PLL clock 14/28MHz. One of them, the line counter clock, can be delayed up to seven steps of 4µseconds by writing to PHD2-PHD0 (R243.2-0). This can be useful in 2H operation to compensate for a substantial phase difference which may exist between chassis deflection pulses and horizontal sync pulses extracted from the video signal.

Figure 12-1. Data-Slicer Block Diagram



DATA SLICER (Continued)

The black level of the CC Video signal is clamped by the Black Level Clamp to a nominal voltage of 2.0V. The clamp is disabled from the rising edge of the OSD generated vertical sync pulse VPLS to the end of the selected "line 21". It is also disabled during the lines used by the Auto Adjusting Logic, inside the Slicing Level Control, to select the Sync optimum slicing level (from line 24 to line 32).

The CC Video signal is fed into the Sync Slicer for the Sync extraction.

Inside this block CC Video is compared with the Sync Slicing Level Voltage (SSLV) set by the Slicing Level Control. Programming RANGE (R243.3) the user can select two sets of Sync and Data slicing levels:

If RANGE = "0" the slicing levels are suitable for a CC Video amplitude of 1Vpp, if RANGE = "1" use input of 2Vpp.

Once the range is set, the fine tuning of SSLV can be set by software or hardware.

If EASSL (R244.5) = "0", SSLV is programmed by the user by writing SSL1-SSL0 (R243.5-4).

If EASSL = "1", SSLV is automatically set by the Auto Adjusting Logic; the user can read the value chosen by the hardware in SSL1-SSL0.

The output of the comparator is digitally filtered to avoid spike propagation, generating CSYNCF.

CSYNCF is fed into the Field and Line21 Detector where its width is compared with a value selected by the user writing PW1-PW0 (R242.7-6) (8, 10, 12, 14µs).

When CSYNCF lasts more than the selected threshold it is interpreted as the beginning of the vertical retrace and the line counter is released from its reset value (0).

With the line counter running, when the counter value matches the value programmed by the user writing LC4-LC0 (R242.4-0), if VPLS was detected (VPDET (R246.6) = "1") the signal LINE21 is activated. With the NTSC standard the user must write the value N-5 in LC to select line N.

Using the clocks provided by the Clock Generator and CSYNCF the current field is recognized. If it is odd the FIELD1 flag (R245.7) is set, otherwise reset. A sequence of odd and even fields keeps the interlace present flag INTLC (R245.6) set. When the Auto Adjusting Logic for data and/or sync is enabled, the user can select in which field to perform the adjustment of the slicing level(s) writing BIASFLD (R242.5). BIASFLD = "1" selects the odd fields.

The CC Video signal is fed into the Data Slicer for the Data extraction. Inside this block CC Video is compared with the Data Slicing Level Voltage (DSLV) set by the Slicing Level Control.

Once the range is set the fine tuning of DSLV can be set by software or hardware.

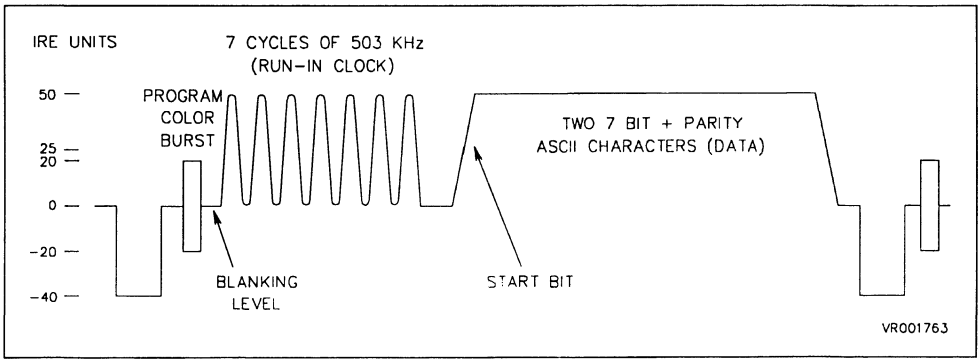
If EADSL (R244.6) = "0" the DSLV is programmed by the user writing DSL1-DSL0 (R243.7-6).

If EADSL = "1" the DSLV is automatically set by the Auto Adjusting Logic; the user can read the value chosen by the hardware in DSL.

The output of the comparator is digitally filtered to avoid spike propagation, generating DSOUTF.

DSOUTF is fed into the Data Processor where it is processed when LINE21 is active. A sequence of seven pulses of 503kHz in phase and with the same amplitude of data (run-in clock) is followed by a zero-zero-one sequence (start bit) and data (2 bytes, seven bits plus odd parity).

Figure 12-2. Closed Caption Data Format



DATA SLICER (Continued)

First the presence of the Run-In Clock (RIC) is checked: if it is not detected within a window of 15µs opened 10µs after the line 21 detection, the NORIC flag (R246.4) is set. If RIC is found the counter RICCNT (R246.2-0) counts the number of RIC pulses; if more than seven pulses are detected the RIC overflow flag RICOVF (R246.3) is set.

A second window of 5µs is opened 25µs after the line 21 detection to check the presence of the Start Bit. If it is not detected the NOSB flag (R246.5) is set and further processing is stopped.

The elapsed time (in µs) between the last falling edge of the RIC and the rising edge of the Start Bit is measured by ZZC field (R245.2-0). A normal value of 4 corresponding to the zero-zero sequence is expected.

Once the Start Bit is detected the two data bytes FD and SD are extracted from DSOUTF. A majority logic algorithm is used to filter data if enabled by T0 (R244.0). The most significant bit of the data is reset if odd parity is found, set otherwise (wrong parity).

When the two bytes are extracted the NEWD flag is set and further processing is stopped. If the NEWD flag was not software reset before a new data acquisition (writing R242), the LOSTD flag is set.

12.1.3 Slicing level control and auto adjusting logic

All the analog levels are generated by the Digital to Analog Converter (DAC).

BLKV is a constant value of about 2.0V, relating to the value of RANGE, used to clamp the video black level.

The other three reference voltages (SSLV, DSLV and PEAKV) have two tuning inputs.

By programming RANGE (R243.3) the user can select two sets of Sync, Data and Peak (see below) slicing levels:

if RANGE = "0" the slicing levels are more suitable for a CC Video amplitude of 1Vpp, if RANGE = "1" the levels are suitable for 2Vpp.

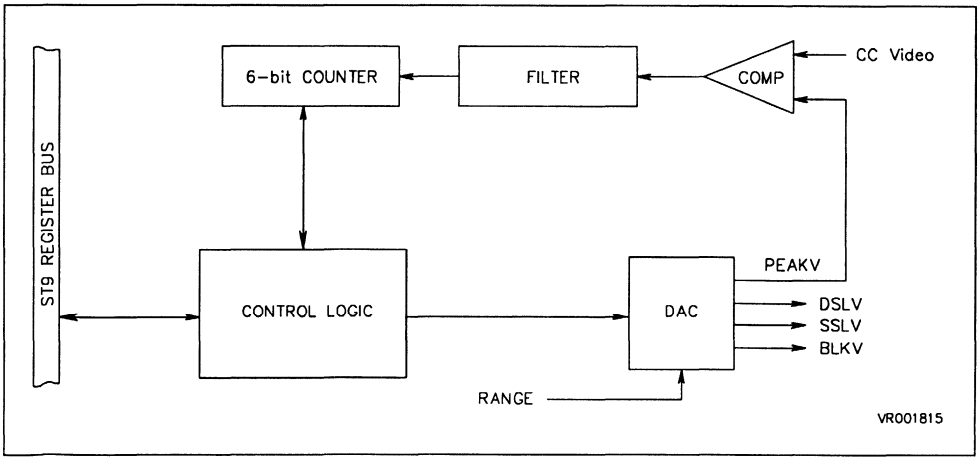
The fine tuning of SSLV can be set by software or hardware.

If EASSL (R244.5) = "0" SSLV is programmed by the user writing SSL1-SSL0 (R243.5-4).

If EASSL = "1" the SSLV is automatically set by the Auto Adjusting Logic (AAL); the user can read the value chosen by the hardware in SSL.

The fine tuning of DSLV can be set by software or hardware.

Figure 12-3. Auto Adjusting Logic



DATA SLICER (Continued)

If EADSL (R244.6) = "0" the DSLV is programmed by the user writing DSL1-DSL0 (R243.7-6).

If EADSL = "1" the DSLV is automatically set by the Auto Adjusting Logic (AAL); the user can read the value chosen by the hardware in DSL.

When switching from EASSL = "1" to EASSL = "0" the last SSL selected by the Auto Adjustment Logic is kept until the first write into SSL. The same applies to DSL.

Read/Write SSL

EASSL	Read SSL	Write SSL
0	from SSL	to SSL
1	from AAL	to SSL

Note: This table applies to EADSL and DSL.

To select automatically the best sync and data slicing level the Auto Adjusting Logic inside the Control Logic uses COMP, FILTER and COUNTER to detect the sync and data peak levels.

Depending on the value of the Auto Adjustment-STATUS (R247.2-0) and RANGE, PEAKV loops in eight frames through eight levels (4 for sync and 4 for data), starting for each search (sync/data) from the absolute maximum value.

The output of COMP, filtered to avoid spike propagation, is used to enable the clock of a 6-bit Counter, enabled by SYNCBIAS during the sync slicing level search and by RICBIAS during the data slicing level search.

During the sync (data) slicing level search, the first time that at least one of the two most significant bits (CNT) of the counter is set, the two less significant bits of Auto Adjustment STATUS are saved and copied at the end of the search loop into SSL_R (DSL_R) if EASSL (EADSL) = "1".

If for any reason the value chosen for sync slicing causes a loss of sync, the absolute minimum value is restored at the end of the search loop.

This works in the background without using any CPU time also if no automatic update is allowed (EASSL = EADSL = "0"). The user can monitor the loop search looking at the value of STATUS and CNT in R247 and select by software which slicing values to use.

Register Description

FD R240 (F0h) page 2B Read-only
First Data Register

Reset value: 1XXX XXXXb (XXh)

7							0
PEFD	FD.6	FD.5	FD.4	FD.3	FD.2	FD.1	FD.0

b7 = **PEFD:** *Parity Error First Data.* Set when wrong (even) parity is detected when sampling first data. Set writing the First Control Register.

b6-b0 = **FD6-FD0:** *First data byte.*

SD R241 (F1h) page 2B Read-only
Second Data Register

Reset value: 1XXX XXXXb (XXh)

7							0
PESD	SD.6	SD.5	SD.4	SD.3	SD.2	SD.1	SD.0

b7 = **PESD:** *Parity Error Second Data.* Set when wrong (even) parity is detected when sampling second data. Set writing the First Control Register.

b6-b0 = **SD6-SD0:** *Second data byte.*

C1 R242 (F2h) page 2B Read/Write
First Control Register

Reset value: 1111 1111b (FFh)

7							0
PW1	PW0	BIASFLD	LC.4	LC.3	LC.2	LC.1	LC.0

b7-b6 = **PW1-PW0:** *Threshold of vertical sync pulse detector.* The horizontal sync extracted from the Composite Video Input (CC Video) is identified as a vertical sync pulse if lasts more than the threshold selected according to the table:

PW1	PW0	Threshold
0	0	8µs
0	1	10µs
1	0	12µs
1	1	14µs

b5 = **BIASFLD:** *Bias Field Select* The automatic search of the optimum Data Slicing Level and Sync Slicing Level is performed when FIELD1 (R245.7) matches the value of this bit.

DATA SLICER (Continued)

b4-b0 = **LC4-LC0:** *Closed-caption Line Selection.* The Data Slicer current line counter is compared with LC value; it is released from the reset state ("0") when the first vertical sync pulse (see PW1-PW0) is recognized.

With NTSC standard, to select line "N" the LC value will be:

LC = N - 5

Line "N" will be selected in both fields and the FIELD1 flag (R245.7) will be set during field 1.

C2 R243 (F3h) page 2B Read/Write
Second Control Register

Reset value: 0000 0000b (00h)

7							0
DSL1	DSL0	SSL1	SSL0	RANGE	PHD2	PHD1	PHD0

b7-b6 = **DSL1-DSL0:** *Data slicing Level select.* Bits setting the Data Slicing Level (DSLV), according to the following table (see RANGE):

DSL	**DSLV** (RANGE = "1")	**DSLV** (RANGE = "0")
00	(1 + 0.22V) BL1	(1 + 0.11V) BL0
01	(1 + 0.18V) BL1	(1 + 0.09V) BL0
10	(1 + 0.14V) BL1	(1 + 0.07V) BL0
11	(1 + 0.10V) BL1	(1 + 0.05V) BL0

Notes.

BL1=Black Level if RANGE=1; BL1(T)=BL1(25)+6.5mV x (T-25)
BL0=Black Level if RANGE=0; BL0(T)=BL0(25)+6.5mV x (T-25)
BLX(25) voltage range: 1.9V - 2.2V
Tolerance for factors: 1%

When EADSL (R244.6) = "0" the DSLV automatic adjustment is disabled. DSLV is software programmed, by writing DSL1-DSL0.

When EADSL = "1" the DSLV automatic adjustment is enabled. Writing DSL1-DSL0 has no effect. By reading DSL1-DSL0 the software can monitor the value selected by the Auto Adjustment Logic.

When switching from EADSL = "1" to EADSL = "0" the last value selected by the Auto Adjustment Logic will remain valid until the first write into R243.

b5-b4 = **SSL1-SSL0:** *Sync slicing Level select.* Bits setting the Sync Slicing Level (SSLV), according to the following table (see RANGE):

SSL	**SSLV** (RANGE = "1")	**SSLV** (RANGE = "0")
00	(1 - 0.20V) BL1	(1 - 0.10V) BL0
01	(1 - 0.17V) BL1	(1 - 0.08V) BL0
10	(1 - 0.14V) BL1	(1 - 0.07V) BL0
11	(1 - 0.11V) BL1	(1 - 0.06V) BL0

Notes:

BL1=Black Level if RANGE=1; BL1(T)=BL1(25)+6.5mV x (T-25)
BL0=Black Level if RANGE=0; BL0(T)=BL0(25)+6.5mV x (T-25)
BLX(25) voltage range: 1.9V - 2.2V
Tolerance for factors: 1%

When EASSL (R244.5) = "0" the SSLV automatic adjustment is disabled. SSLV is software programmed, by writing SSL1-SSL0.

When EASSL = "1" the SSLV automatic adjustment is enabled. Writing SSL1-SSL0 has no effect. Reading SSL1-SSL0 the software can monitor the value selected by the Auto Adjustment Logic.

When switching from EASSL = "1" to EASSL = "0" the last value selected by the Auto Adjustment Logic will remain valid until the first write into R243.

b3 = **RANGE:** *Slicing Voltage Range Select.* This bit selects the voltage range used for slicing both data and sync.

RANGE = "0" selects a slicing voltage range optimized for a 1 Vpp signal.

RANGE = "1" selects a slicing voltage range optimized for a 2 Vpp signal.

b2-b0 = **PHD2-PHD0:** *Phase Delay.*

These bits specify the delay added to the current line counter clock in the line detection circuitry.

A delay can be specified in 2H mode operation to compensate for a substantial phase difference which may exist between chassis deflection pulses and horizontal sync pulses extracted from the video signal.

Table 12-1. Delay Value

PHD2	PHD1	PHD0	**Phase Delay** (µs)
0	0	0	no delay
0	0	1	4
0	1	0	8
0	1	1	12
1	0	0	16
1	0	1	20
1	1	0	24
1	1	1	28

DATA SLICER (Continued)

C3 R244 (F4h) page 2B Read/Write
Third Control Register

Reset value: 0000 0000b (00h)

7							0
EDS	EADSL	EASSL	T.4	T.3	T.2	T.1	MAJE

b7 = **EDS:** *Enable Data Slicer.* EDS is active high to enable the Data Slicer.
When EDS = "0" the Data Slicer is disabled.

The clock of the peripheral is stopped and the following bits are forced in their reset state:

- R245: FIELD1, INTLC, LOSTD, NEWD, ZZCNT;
- R246: CSYNC, NOSB, NORIC, RICOVF, RICCNT;
- R247: CNTOVF, CNT, STATUS.

b6 = **EADSL:** *Enable Automatic Data Slicing Level.* Active high to enable the automatic adjustment for the Data Slicer.

b5 = **EASSL:** *Enable Automatic Sync Slicing Level.* Active high to enable the automatic adjustment for the Sync Slicer.

b4-b1 = **T.4-T.1:**Reserved bits, must be held low ("0").

b0 = **MAJE:** *Majority Logic Enable.* Active low to enable the Majority Logic(ML). Each of the 16bits of the closed Caption data is sampled 14 times to avoid error by noise. After the 14 samples are taken, the last value is discarded and the majority value among the 13 sampled values is taken as the value of the sampled bit.

e.g. If 7 sampled values are "1", and 6 are "0", the majority result is taken as "1".

The majority routine repeats for each bit of the data and takes 2µs for each bit.

If the Majority Logic is disabled each bit of the Closed Caption data is sampled only once at the middle of each bit (1µs after each bit is detected). This sampled value is taken as the result of the bit.

M1 R245 (F5h) page 2B Read-only
First Monitor Register

Reset value: 0000 0000b (00h)

7							0
FIELD1	INTLC	LOSTD	NEWD	0	ZZC2	ZZC1	ZZC0

b7 = **FIELD1:** *Field 1 Flag.* Set during field1.
Reset when EDS (R244.7) = "0".

b6 = **INTLC:** *Interlace Flag.* Set when fields are interlaced.
Reset when EDS (R244.7) = "0".

b5 = **LOSTD:** *Lost Data Flag.* NEWD is loaded into LOSTD at the beginning of the selected line.
Reset when EDS (R244.7) = "0". Reset writing the First Control Register.

b4 = **NEWD:** *New Data Available.* Set at the end of a new data pair acquisition.
Reset when EDS (R244.7) = "0".

Reset writing the First Control Register.

b3-b0 = **ZZC2-ZZC0:** *Zero Zero Count.* Measure of the "00" sequence before the Start Bit (in microseconds).
Reset when EDS (R244.7) = "0" .

Reset at the beginning of the selected line.

M2 R246 (F6h) page 2B Read-only
Second Monitor Register

Reset value: 0x01 0000b (x0h)

7							0
CSYNC	VPDET	NOSB	NORIC	RICOVF	RICC2	RICC1	RICC0

b7 = **CSYNC:** *Composite Sync Detected.* Set during Horizontal Sync, extracted from the Video Composite Input.
Reset when EDS (R244.7) = "0" .

b6 = **VPDET:** *Vertical Sync Detected.* Set when Vertical Pulse is detected.
Reset writing the First Control Register.

Note: To garantee a correct detection of line 21 the reset of VPDET must occur after line 37 (NTSC standard).

DATA SLICER (Continued)

b5 = **NOSB:** *No Start Bit.* Set when No Start Bit is detected.
Reset when EDS (R244.7) = "0" .

Reset at the beginning of the selected line.

b4 = **NORIC:** *No Run In Clock.* Set when No Run In Clock is detected.
Set when EDS (R244.7) = "0" .

Set at the beginning of the selected line.

b3 = **RICOVF:** *RIC Overflow.* Set when more than seven Run In Clock pulses are detected.
Reset when EDS (R244.7) = "0" .

Reset at the beginning of the selected line.

b2-b0 = **RICC2-RICC0:** *RIC Counter.* Run In Clock detected pulses.
Reset when EDS (R244.7) = "0".

Reset at the beginning of the selected line.

M3 R247 (F7h) page 2B Read-only
Third Monitor Register

Reset value: 0000 0000b (00h)

7							0
CNTOVF	CNT1	CNT0	0	0	STAT.2	STAT.1	STAT 0

b7 = **CNTOVF:** *CNT Overflow.* Set when CNT overflows.
Reset when STATUS changes.

Reset when EDS (R244.7) = "0".

b6-b5 = **CNT1-CNT0:** *Auto Adjustment Counter.* Two most significant bits of the 6-bit counter used by the Auto Adjustment Logic to detect the SYNC or DATA peak level.
Reset when STATUS changes.

Reset when EDS (R244.7) = "0".

b2-b0 = **STAT.2-STAT.0:** *Auto Adjustment Status.* Status of the Auto Adjustment Logic.

This field monitors the current Peak Level used by the Auto Adjusting Logic for the SYNC or DATA peak level detection.

The voltage range used for the peak level detection is software controlled by RANGE (R243.3).

STATUS	Peak Level (RANGE = "1")	Peak Level (RANGE = "0")
100	(1 + 0.36V) BL1	(1 + 0.18V) BL0
101	(1 + 0.29V) BL1	(1 + 0.14V) BL0
110	(1 + 0.22V) BL1	(1 + 0.11V) BL0
111	(1 + 0.14V) BL1	(1 + 0.07V) BL0
011	(1 - 0.17V) BL1	(1 - 0.08V) BL0
010	(1 - 0.20V) BL1	(1 - 0.10V) BL0
001	(1 - 0.23V) BL1	(1 - 0.11V) BL0
000	(1 - 0.26V) BL1	(1 - 0.13V) BL0

Notes.

BL1=Black Level if RANGE=1; BL1(T)=BL1(25)+6.5mV x (T-25)
BL0=Black Level if RANGE=0; BL0(T)=BL0(25)+6.5mV x (T-25)
BLX(25) voltage range: 1.9V - 2.2V
Tolerance for factors: 1%

STATUS changes every frame (2 fields).

Reset when EDS (R244.7) = "00".

25 IRE (B.L+0.350 V)

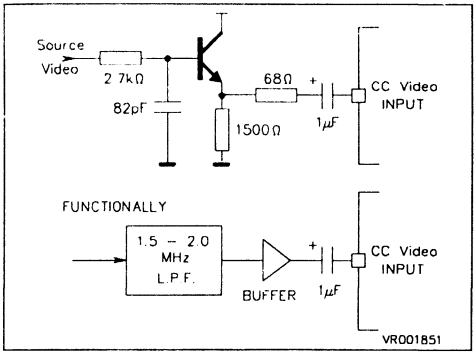
12.1.4 Data Slicer Interface

The Data Slicer accepts video signals for data extraction through the CC Video pin. The composite video signal is applied to this pin AC is to be coupled with a 1µF capacitor.

On Screen Display inputs. The following signals, generated by the On Screen Display, must be available for the Data Slicer to operate:

- HPLS: HSYNC pin after internal polarity adjustment; positive pulse.
- VPLS: VSYNC pin after internal polarity adjustment; positive pulse.
- 14/28 MHz: PLL clock (14 MHz in 1H mode)
- H1/H2: Bit 2 of OSD Enable Register, selecting 1H or 2H mode.

Typical External Circuitry



NOTES

13 PWM GENERATOR

13.1 INTRODUCTION

The ST9 PWM multiple output Unit is an 8 independant channel Pulse Width Modulated (PWM) signal generator allowing the digital generation of analog outputs when used with an external filtering network.

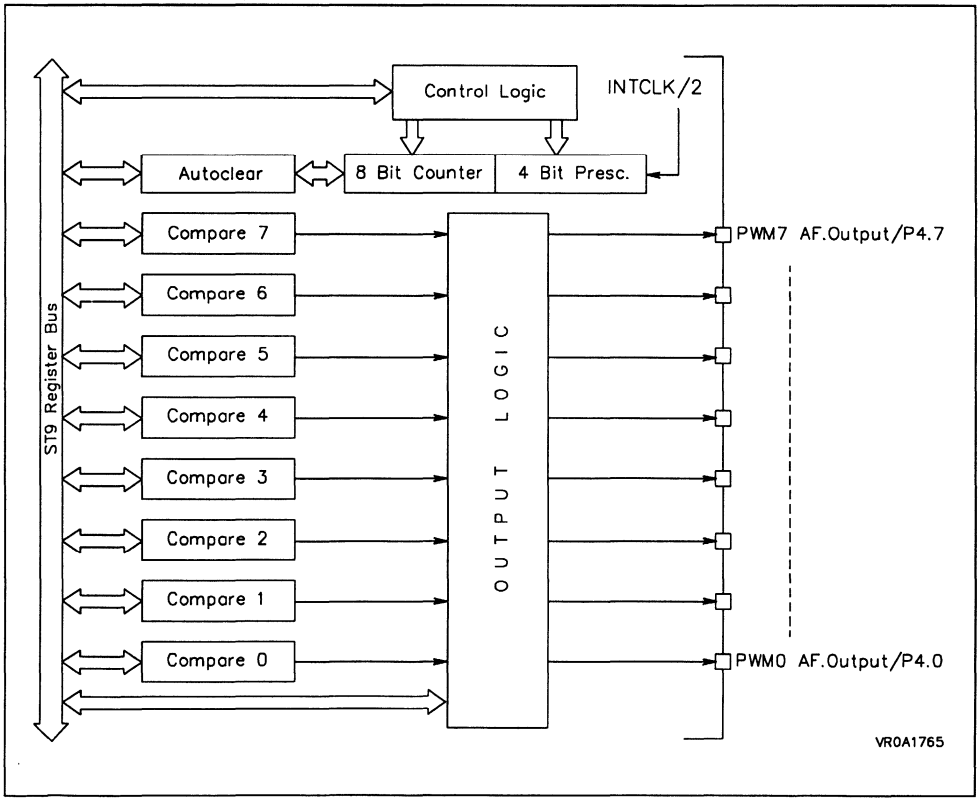
The unit is based around an 8-bit counter which is driven by a programmable 4-bit prescaler, with an input clock signal equal to the internal clock INTCLK divided by two. At 12MHz Internal clock, using the full 8 bit resolution, a frequency range from 1465Hz up to 23437Hz can be achieved.

Higher frequencies, with lower resolution, can be achieved by using the autoclear register. As an example, at 12MHz Internal clock, a maximum PWM repetition rate of 93750Hz can be reached with 6 bit resolution.

Up to 8 PWM outputs can be selected as Alternate Functions of an I/O port. Each output bit is independantly controlled by a separate Comparison Register. When the value programmed into the Compare Register and the counter value are equal, the corresponding output bit is set. The output bit is reset by a counter clear (by overflow or autoclear), generating the variable PWM signal.

Each output bit can also be complemented or disabled under software control.

Figure 13-1. PWM Block Diagram



PWM GENERATOR (Continued)

13.1.1 Register Mapping

The PWM Unit has 13 registers mapped in page 59 (3Bh), and defined as following:

Register Address	Register	Function
F0	CP0	Ch. 0 Compare Register
F1	CP1	Ch. 1 Compare Register
F2	CP2	Ch. 2 Compare Register
F3	CP3	Ch. 3 Compare Register
F4	CP4	Ch. 4 Compare Register
F5	CP5	Ch. 5 Compare Register
F6	CP6	Ch. 6 Compare Register
F7	CP7	Ch. 7 Compare Register
F8	ACR	Autoclear Register
F9	CRR	Counter Read Register
FA	PCTLR	Prescaler/Reload Reg.
FB	OCPLR	Output Complement Reg.
FC	OER	Output Enable Register
FD-FF	—	Reserved

13.1.2 Register Description

OE R252 (FCh) Page 3B Read/Write
Output Enable Register

Reset Value: 0000 0000b (00h)

7							0
OE.7	OE.6	OE.5	OE.4	OE.3	OE.2	OE.1	OE.0

Each bit of this register locks the corresponding PWM output to the logic level "1" when the bit is reset to "0". This allows the bit to be used for normal I/O functions or other alternate functions (if available).

When the corresponding bit is set to "1", the PWM output function is enabled.

For example, writing 03h into the OE Register will enable only the PWM outputs 0 and 1, while the outputs 2, 3, 4, 5, 6 and 7 will be forced to the "1" logic level.

b7 = **OE.7:** *Output Enable PWM Output 7.*

b6 = **OE.6:** *Output Enable PWM Output 6.*

b5 = **OE.5:** *Output Enable PWM Output 5.*

b4 = **OE.4:** *Output Enable PWM Output 4.*

b3 = **OE.3:** *Output Enable PWM Output 3.*

b2 = **OE.2:** *Output Enable PWM Output 2.*

b1 = **OE.1:** *Output Enable PWM Output 1.*

b0 = **OE.0:** *Output Enable PWM Output 0.*

OCPL R251 (FBh) Page 3B Read/Write
Output Complement Register

Reset Value: 0000 0000b (00h)

7							0
OCPL.7	OCPL.6	OCPL.5	OCPL.4	OCPL.3	OCPL.2	OCPL.1	OCPL.0

This register allows the PWM output level to be complemented on an individual bit basis.

In the default mode (reset configuration), each comparison true between a Compare register and the counter has the effect to set the corresponding output.

At counter clear (either by autoclear comparison true, software clear or overflow when in free running mode), all the outputs are cleared.

By setting each individual bit (OCPL.x) in this register, the logic value of the corresponding output will be inverted (i.e. reset on comparison true and set on counter clear).

For example: When set to "1", OCPL.1 Bit complements the PWM output 1.

b7 = **OCPL.7:** *Complement PWM Output 7.*

b6 = **OCPL.6:** *Complement PWM Output 6.*

b5 = **OCPL.5:** *Complement PWM Output 5.*

b4 = **OCPL.4:** *Complement PWM Output 4.*

b3 = **OCPL.3:** *Complement PWM Output 3.*

b2 = **OCPL.2:** *Complement PWM Output 2.*

b1 = **OCPL.1:** *Complement PWM Output 1.*

b0 = **OCPL.0:** *Complement PWM Output 0.*

PCTL R250 (FAh) Page 3B Read/Write
Prescaler and Control Register

Reset Value: 0000 1100b (0Ch)

7							0
PR3	PR2	PR1	PR0	1	1	CLR	CE

b7-b4 = **PR3-0** *PWM Prescaler value.* These bits hold the Prescaler preset value. This is reloaded into the 4-bit prescaler whenever the prescaler (DOWN Counter) reaches the value 0, so determining the 8-bit Counter count frequency. The value 0 corresponds to the maximum counter frequency which is the ST9 Internal Clock divided by 2 (6Mhz at 12Mhz INTCLK). The value 1 corresponds to the maximum frequency divided by 2. The value F corresponds the maximum frequency divided by 16.

The reset condition initializes the Prescaler to the Maximum Counter frequency.

b3-b2 = set to "1"

PWM GENERATOR (Continued)

b1 = **CLR Bit:** *Counter Clear.* This bit, when set, allows both to clear the counter, and to reload the prescaler. The effect is also to clear the PWM output. It returns "0" if read.

b0 = **CE:** *Counter Enable.* This bit enables the counter and the prescaler when set to "1". It stops both when reset without affecting their current value, allowing the count to be suspended and then restarted by software "on fly".

C R249 (F9h) Page 3B Read Only
Counter Register

Reset Value: 0000 0000b (00h)

7							0
CR.7	CR.6	CR.5	CR.4	CR.3	CR.2	CR.1	CR.0

This read only register returns the current counter value when read.

The 8 bit Counter is initialized to 00h at reset, and is a free running UP counter.

b7-b0 = **CR.7-CR.0:** *Current Counter Value.*

AC R248 (F8h) Page 3B Read/Write
AutoClear Register

Reset Value: 1111 1111b (FFh)

7							0
AC.7	AC.6	AC.5	AC.4	AC.3	AC.2	AC.1	AC.0

This register behaves exactly as a 9th compare Register, but its effect is to clear the 8 bit counter, so causing the desired PWM repetition rate.

The reset condition generates the free running mode. So, FFh means count by 256.

b7-b0 = **AC.7-AC.0:** *Autoclear Count Value.*

If Autoclear register = FFh , when 00 is written to the Compare Register, the PWM output is always set except for the last clock count (255 set and 1 reset; the converse when the output is complemented).
Whenever the autoclear register content is less than 255, the PWM output is set for a number of clock counts equal to that content.

Writing FFh to the compare register resets the output always (sets if the output is complemented). This is also the case when the compare register constant is equal to the autoclear register content.

Figure 13-2a. PWM Action on Autoclear

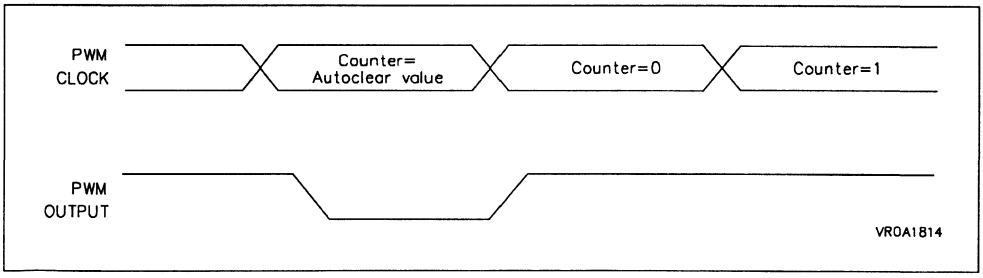
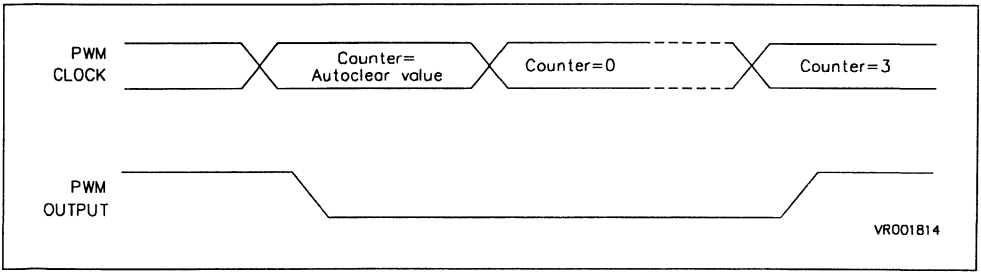


Figure 13-2b. PWM Action on PWM Compare Register = 3 (no complement)



PWM GENERATOR (Continued)

For example, when 03 is written, the output is reset when the counter reaches the autoclear register value and set when it reaches the compare value (after 4 clock counts). Vice-versa if the output is complemented.

The PWM mark/space ratio will remain constant until changed by software.

CM7 R247 (F7h) Page 3B Read/Write
Compare Register 7

Reset Value: 0000 0000b (00h)

7							0
CM7.7	CM7.6	CM7.5	CM7.4	CM7.3	CM7.2	CM7.1	CM7.0

This is the compare register controlling PWM output 7. When the programmed content is equal to the counter content, a SET operation is performed on PWM output 7 (if the output has not been complemented or disabled).

b7-b0 = **CM7.7-CM7.0:** *PWM Compare value Channel 7.*

CM6 R246 (F6h) Page 3B Read/Write
Compare Register 6

Reset Value: 0000 0000b (00h)

7							0
CM6.7	CM6.6	CM6.5	CM6.4	CM6.3	CM6.2	CM6.1	CM6.0

This is the compare register controlling PWM output 6.

CM5 R245 (F5h) Page 3B Read/Write
Compare Register 5

Reset Value: 0000 0000b (00h)

7							0
CM5.7	CM5.6	CM5.5	CM5.4	CM5.3	CM5.2	CM5.1	CM5.0

This is the compare register controlling PWM output 5.

CM4 R244 (F4h) Page 3B Read/Write
Compare Register 4

Reset Value: 0000 0000b (00h)

7							0
CM4.7	CM4.6	CM4.5	CM4.4	CM4.3	CM4.2	CM4.1	CM4.0

This is the compare register controlling PWM output 4.

CM3 R243 (F3h) Page 3B Read/Write
Compare Register 3

Reset Value: 0000 0000b (00h)

7							0
CM3.7	CM3.6	CM3.5	CM3.4	CM3.3	CM3.2	CM3.1	CM3.0

This is the compare register controlling PWM output 3.

CM2 R242 (F2h) Page 3B Read/Write
Compare Register 2

Reset Value: 0000 0000b (00h)

7							0
CM2.7	CM2.6	CM2.5	CM2.4	CM2.3	CM2.2	CM2.1	CM2.0

This is the compare register controlling PWM output 2.

CM1 R241 (F1h) Page 3B Read/Write
Compare Register 1

Reset Value: 0000 0000b (00h)

7							0
CM1.7	CM1.6	CM1.5	CM1.4	CM1.3	CM1.2	CM1.1	CM1.0

This is the compare register controlling PWM output 1.

CM0 R240 (F0h) Page 3B Read/Write
Compare Register 0

Reset Value: 0000 0000b (00h)

7							0
CM0.7	CM0.6	CM0.5	CM0.4	CM0.3	CM0.2	CM0.1	CM0.0

This is the compare register controlling PWM output 0.

14 IFP A/D CONVERTER

14.1 GENERAL DESCRIPTION

The IFP Analog to Digital converter peripheral can convert up to 3 analog channels using the successive approximation conversion technique to a resolution of 8 bits ±2LSB. It includes four analog fully differential comparators blocks, AC coupled, plus Sample and Hold logic and a reference generator.

The internal reference (DAC) is based on the use of a binary-ratioed capacitor array. This technique provides good monotonicity by using the same ratioed capacitors as the sample capacitor.

The Single Conversion time is 5.75µs (INTCLK = 12MHz) including a Sample time of 1.5µs (typical). The 8-bit conversion uses 69 INTCLK periods for the conversion.

A continuous conversion of the selected channel is performed in Continuous mode, while in the Single mode the selected channel is only once converted and the logic then waits for a new hardware or software restart.

The start conversion event can be produced by software, by writing the START/STOP bit of the Control Logic Register (CLR), or by external hardware, by applying an external trigger on the EXTRG (negative edge sensitive).

An interrupt may be generated at the End of conversion i.e. at any time the STR bit performs a high to low transition. This interrupt is connected to the INT3 channel and takes the priority and vector associated to the External Interrupt Channel INTB1.

The Data register holds the converted value data (in single or continuous mode).

A Power Down programmable bit sets the A/D converter analog section to a minimum consumption idle status.

14.1.1 Operation Modes

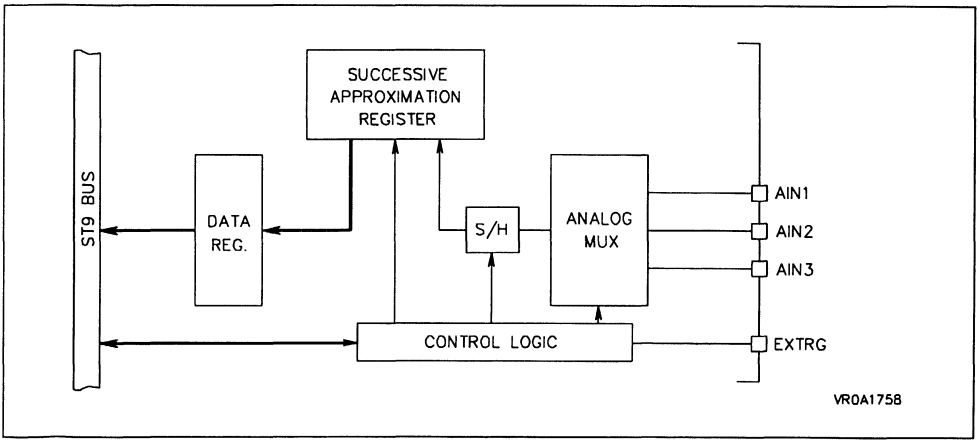
The two main operation modes, SINGLE and CONTINUOUS, can be selected writing "0" (reset value) or "1" into the CONT bit of the Control Logic Register:

In **Single Mode** (CONT = "0" into CLR) the STR bit is forced to "0" after the end of channel `i` conversion (`i` = 1, 2 or 3); the A/D then waits for a new start event.

This mode is useful when a set of signals must be sampled at a fixed frequency imposed by an ST9 timer unit or an external generator (through the alternate synchronization feature).

A simple software routine monitoring the STR bit can be used to save the current value before a new conversion end (so to create signal samples table within internal memory or Register File).

Figure 14-1. IFP A/D Converter Block Diagram



IFP A/D CONVERTER (Continued)

In **Continuous Mode** (CONT = "1" into CLR) a continuous conversion flow is entered by a start event. After the end of conversion of channel i the conversion takes place again on channel i, repeating until the STR bit is reset by software.

At the end of conversion of channel i, the Data Register (DTR) is reloaded with the new conversion result and the previous value is lost.

The **Alternate Synchronization Feature** is available in both single and continuous operation modes. The external EXTRG signal can be used to synchronize the conversion start with a trigger pulse. The external events can be masked by programming the TRG bit of the Control Logic Register.

The effect of the alternate synchronization is to hardware set the STR bit, which is hardware reset at the end of each conversion only in SINGLE mode. In CONTINUOUS mode all trigger pulses, following the first one, are ignored.

The synchronization source must have a clock cycle minimum length of 83ns (at INTCLK = 12Mhz) minimum length, and a period greater (in Single mode) than the time occurring for a conversion.

If a trigger occurs when the STR bit is still "1" (conversion still in progress), it is ignored.

Note: Before enabling any A/D operation mode, it is necessary to set the POW bit of the Control Logic Register to "1" to bias the analog section of the converter at least 60µs before the first conversion; otherwise the converter functionality is locked. Setting POW to "0" is useful when the IFP A/D is not used to reduce the total power consumption. This is the reset configuration and is also automatically entered when the ST9 is in the HALT state (HALT instruction executed).

14.1.2 Register Mapping

The IFP A/D peripheral has 2 dedicated registers, mapped in the ST9 Register File page 62 (3Eh) as follows:

Register Address	Page 62 (3Eh)
F0	DTR
F1	CLR
F2-FF	Reserved

14.1.3 Register Description

DTR R240 (F0h) Page 3E Read/Write
Channel i Data Register

Reset value: undefined

7							0
R.7	R.6	R.5	R.4	R.3	R.2	R.1	R.0

The results of the conversion of the selected channel is stored into the 8 bit DTR, which is reloaded with a new value every time a conversion ends.

b7-b0 = **R7-R0:** *Channel i* conversion result.

Note: Only the most significant 6 bits are characterised when the ST9 is operating during conversion (accuracy is typically 8-bit ± 2LSB). When the ST9 is in WFI mode the accuracy is increased.

CLR R241 (F1h) Page 3E Read/Write

IFP A/D CONVERTER (Continued)

Control Logic Register

Reset value: 0000 0000 (00h)

7							0
C2	C1	C0	FS	TRG	POW	CONT	STR

This 8 bit register manages the A/D logic operations. Any write operation to it will cause the current conversion to be aborted and the logic to be re-initialized to the starting configuration.

b7-b5 = **C2-C0:** *Channel Address.*

A "1" in bit Ci selects channel i conversion as follows:

C2	C1	C0	Channel Enabled
0	0	0	No channel enabled
0	0	1	1
0	1	0	2
1	0	0	3

Warning: If more than one channel is simultaneously enabled (the remaining C2-C1 combinations), the selected channels are connected together by hardware.

b4 = **FS:** *Fast/Slow.* A logical "1" level enables a division by two of the conversion frequency, doubling the sampling time.

FS = "0" (fast mode) Single conversion time: 5.75µs at INTCLK = 12MHz

FS = "1" (slow mode) Single conversion time: 11.5µs at INTCLK = 12MHz

b3 = **TRG:** *External Trigger Enable.* A logical level "0" on this bit disables the External Trigger applied on the EXTRG pin. When TRG is set to "1", a negative (falling) edge on TRG writes a "1" into the STR bit, enabling start of conversion.

b2 = **POW:** *Power Enable.* A logical "1" level enables the A/D logic and analog circuitry. A logical level "0" disables all power consuming logic within the IFP A/D.

b1 = **CONT:** *Continuous/Single Mode Select.* A logical level "1" sets the converter in 'Continuous Mode', thus allowing a continuous flow of A/D conversions on the selected channel, starting when the STR bit is set.

A logical level "0" sets the 'Single mode'; after the current conversion end, the STR bit is hardware reset and the converter logic is put in a wait status. To start another conversion, the STR bit has to be set by software or hardware.

b0 = **STR:** *Start/Stop.* A "1" enables the start of conversion of channel i; a logical level "0" stops the converter until a new "1" is written (by hardware, when the A/D is synchronized with an external trigger, or by software).

NOTES

REGISTER MAP

2 CORE ARCHITECTURE

CICR	**R230**	(E6h)	System	Read/Write	Central Interrupt Control Register	11
FLAGR	**R231**	(E7h)	System	Read/Write	Flag Register	12
RP0	**R232**	(E8h)	System	Read/Write	Register Pointer 0	13
RP1	**R233**	(E9h)	System	Read/Write	Register Pointer 1	13
PPR	**R234**	(EAh)	System	Read/Write	Page Pointer Register	15
MODER	**R235**	(EBh)	System	Read/Write	Mode Register	15
USP	**R236**	(ECh)	System	Read/Write	User Stack Pointer High Byte	17
USP	**R237**	(EDh)	System	Read/Write	User Stack Pointer Low Byte	17
SSP	**R238**	(EEh)	System	Read/Write	System Stack Pointer High Byte	17
SSP	**R239**	(EFh)	System	Read/Write	System Stack Pointer Low Byte	17

4 INTERRUPTS

CICR	**R230**	(E6h)	System	Read/Write	Central Interrupt Control Register	36
EITR	**R242**	(F2h)	Page 0	Read/Write	External Interrupt Trigger Event Register	36
EIPR	**R243**	(F3h)	Page 0	Read/Write	External Interrupt Pending Register	36
EIMR	**R244**	(F4h)	Page 0	Read/Write	External Interrupt Mask-bit Register	37
EIPLR	**R245**	(F5h)	Page 0	Read/Write	External Interrupt Priority Level Register	37
EIVR	**R246**	(F6h)	Page 0	Read/Write	External Interrupt Vector Register	37
NICR	**R247**	(F7h)	Page 0	Read/Write	Nested Interrupt Control Register	37

5 CLOCK

MODER	**R235**	(EBh)	System	Read/Write	Mode Register	40

8 SERIAL PERIPHERAL INTERFACE

SPIDR	**R253**	(FDh)	Page 0	Read/Write	SPI Data Register	54
SPICR	**R254**	(FEh)	Page 0	Read/Write	SPI Control Register	54
SWAP	**R255**	(FFh)	Page 0	Read/Write	SPI SWAP Control Register	54

9 TIMER/WATCHDOG

WDTHR	**R248**	(F8h)	Page 0	Read/Write	Timer/Watchdog Counter Register, High byte	64
WDTLR	**R249**	(F9h)	Page 0	Read/Write	Timer/Watchdog Counter Register, Low byte.	64
WDTPR	**R250**	(FAh)	Page 0	Read/Write	Timer/Watchdog Prescaler Register	64
WDTCR	**R251**	(FBh)	Page 0	Read/Write	Timer/Watchdog Control Register	64
WCR	**R252**	(FCh)	Page 0	Read/Write	Wait Control Register	65
EIVR	**R246**	(F6h)	Page 0	Read/Write	External Interrupt Vector Register	65

10 SLICE TIMER

STH	**R240**	(F0h)	Page 0B	Read/Write	Counter High-Byte Register	68
STL	**R241**	(F1h)	Page 0B	Read/Write	Counter Low-Byte Register	68
STP	**R242**	(F2h)	Page 0B	Read/Write	Slice Timer Prescaler Register	68
STC	**R243**	(F3h)	Page 0B	Read/Write	Slice Timer Control Register	68

11 ON SCREEN DISPLAY

HDCS	**R244**	(F4h)	Page 2A	Read/Write	Horizontal Delay/Character Size Register	70
AR	**R245**	(F5h)	Page 2A	Read/Write	Active Range Register	71
SL	**R249**	(F9h)	Page 2A	Read-only	Scan Line Register	74
FB	**R250**	(FAh)	Page 2A	Read-only	Flag Bit Register	74
BCOL	**R251**	(FBh)	Page 2A	Read/Write	Border Color Register	74
EN	**R247**	(F7h)	Page 2A	Read/Write	Enable Register	75
EL	**R246**	(F6h)	Page 2A	Read/Write	Event Line Register	75
VD	**R248**	(F8h)	Page 2A	Read/Write	Vertical Delay Register	75

12 DATA SLICER

FD	**R240**	(F0h)	page 2B	Read-only	First Data Register	82
SD	**R241**	(F1h)	page 2B	Read-only	Second Data Register	82
C1	**R242**	(F2h)	page 2B	Read/Write	First Control Register	82
C2	**R243**	(F3h)	page 2B	Read/Write	Second Control Register	83
C3	**R244**	(F4h)	page 2B	Read/Write	Third Control Register	84
M1	**R245**	(F5h)	page 2B	Read-only	First Monitor Register	84
M2	**R246**	(F6h)	page 2B	Read-only	Second Monitor Register	84
M3	**R247**	(F7h)	page 2B	Read-only	Third Monitor Register	85

13 PWM GENERATOR

OE	**R252**	(FCh)	Page 3B	Read/Write	Output Enable Register	88
OCPL	**R251**	(FBh)	Page 3B	Read/Write	Output Complement Register	88
PCTL	**R250**	(FAh)	Page 3B	Read/Write	Prescaler and Control Register	88
C	**R249**	(F9h)	Page 3B	Read Only	Counter Register	89
AC	**R248**	(F8h)	Page 3B	Read/Write	AutoClear Register	89
CM7	**R247**	(F7h)	Page 3B	Read/Write	Compare Register 7	90
CM6	**R246**	(F6h)	Page 3B	Read/Write	Compare Register 6	90
CM5	**R245**	(F5h)	Page 3B	Read/Write	Compare Register 5	90
CM4	**R244**	(F4h)	Page 3B	Read/Write	Compare Register 4	90
CM3	**R243**	(F3h)	Page 3B	Read/Write	Compare Register 3	90
CM2	**R242**	(F2h)	Page 3B	Read/Write	Compare Register 2	90
CM1	**R241**	(F1h)	Page 3B	Read/Write	Compare Register 1	90
CM0	**R240**	(F0h)	Page 3B	Read/Write	Compare Register 0	90

14 IFP A/D CONVERTER

DTR	**R240**	(F0h)	Page 3E	Read/Write	Channel i Data Register	92
CLR	**R241**	(F1h)	Page 3E	Read/Write	Control Logic Register	93

NOTES

15 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	V_{SS} – 0.3 to 7.0	V
AV_{DD}	Analog Supply Voltage	$AV_{DD} \geq V_{SS}$ - 0.3 to 7.0 V_{DD} - 0.3 ≤ AV_{DD} ≤ V_{DD} + 0.3	V
V_I	Input Voltage	V_{SS} – 0.3 to V_{DD} + 0.3	V
V_O	Output Voltage	V_{SS} – 0.3 to V_{DD} + 0.3	V
T_{STG}	Storage Temperature	– 55 to + 150	°C

Note: Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Value		Unit
		Min.	Max.	
T_A	Operating Temperature	0	70	°C
V_{DD}	Operating Supply Voltage	4.5	5.5	V
f_{OSCE}	External Oscillator Frequency		24	MHz
f_{OSCI}	Internal Oscillator Frequency		12	MHz

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5V ± 10% T_A = 0°C to + 70°C, unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IHCK}	Clock Input High Level	External Clock	0.7 V_{DD}			V
V_{ILCK}	Clock Input Low Level	External Clock			0.3 V_{DD}	V
V_{IH}	Input High Level	TTL	2.0			V
		CMOS	0.7 V_{DD}			V
V_{IL}	Input Low Level	TTL			0.8	V
		CMOS			0.3 V_{DD}	V
V_{IHRS}	$\overline{\text{RESET}}$ Input High Level		0.7 V_{DD}			V
V_{ILRS}	$\overline{\text{RESET}}$ Input Low Level				0.3 V_{DD}	V
V_{HYRS}	$\overline{\text{RESET}}$ Input Hysteresis		0.3			V
V_{IHY}	P2.0, P2.1 Input Hysteresis		0.9		1.5	V
V_{HYHV}	HSYNC/VSYNC Hysteresis		0.5			V
V_{OH}	Output High Level	Push Pull, Iload = – 0.8mA	VDD – 0.8			V
V_{OL}	Output Low Level	Push Pull or Open Drain, Iload = 2mA			0.4	V
I_{WPU}	Weak Pull-up Current	Bidirectional Weak Pull-up, V_{OL} = 0V	– 80	– 200	– 420	µA
I_{OH}	High Voltage Open Drain Output Leakage Current	V_{OH} =12V			+ 30	
I_{LKIO}	I/O Pin Input Leakage	Input/Tri-State, 0V < V_{IN} < V_{DD}	– 10		+ 10	µA
I_{LKRS}	$\overline{\text{RESET}}$ Pin Input Leakage	0V < V_{IN} < V_{DD}	– 30		+ 30	µA
I_{LKAD}	A/D Pin Input Leakage	Alternate Function, Open Drain, 0V < V_{IN} < V_{DD}	– 3		+ 3	µA
I_{LKOS}	OSCIN Pin Input Leakage	0V < V_{IN} < V_{DD}	– 10		+ 10	µA

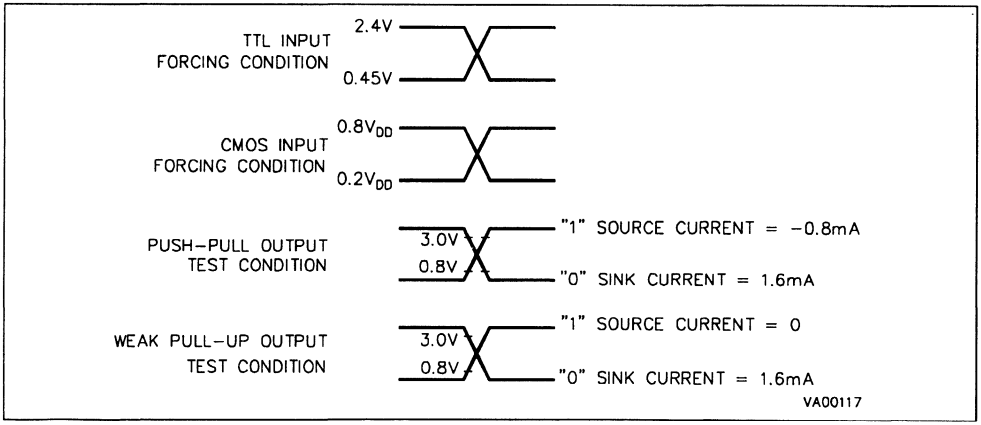
Note:
1. All I/O Ports are configured in Bidirectional Weak Pull-up Mode with no DC load, External Clock pin (OSCIN) is driven by square wave external clock. No peripheral working.

AC ELECTRICAL CHARACTERISTICS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
I_{DD}	Run Mode Current no CPUCLK prescale, CLOCK divide by 2	24MHz, Note 1		40	70	mA
I_{DP2}	Run Mode Current Prescale by 2 CLOCK divide by 2	24MHz, Note 1		19	40	mA
I_{WFI}	WFI Mode Current no CPUCLK prescale, CLOCK divide by 2	24MHz, Note 1		15	20	mA
I_{HALT}	HALT Mode Current	24MHz, Note 1		50	100	µA
I_{LPR}	Low Power Reset Current, Note 2			TBD	TBD	µA

Notes:
1. All I/O Ports are configured in Bidirectional Weak Pull-up Mode with no DC load, External Clock pin (OSCIN) is driven by square wave external clock. No peripheral working.
2. All I/O Port connected to VDD by 10kΩPull-up, External Clock pin (OSCIN) Stopped.

AC TEST CONDITIONS



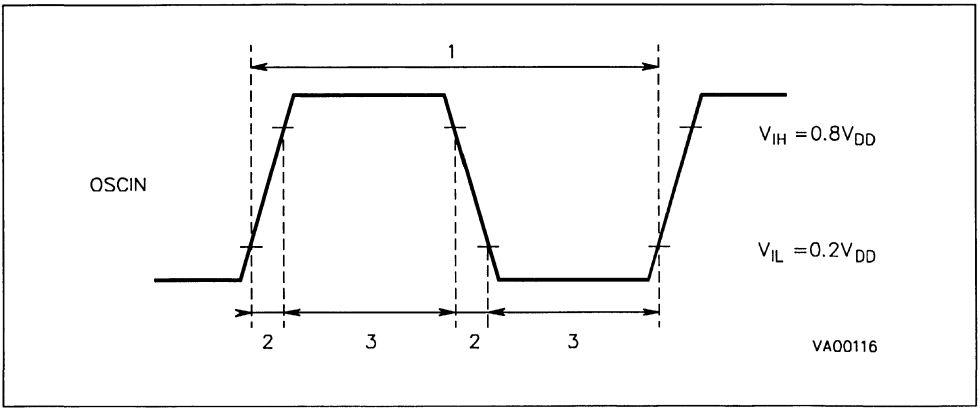
CLOCK TIMING TABLE

(V_{DD} = 5V ± 10%, T_A = − 0°C to + 70°C, INTCLK = 12MHz, unless otherwise specified)

N°	Symbol	Parameter	Value		Unit	Note
			Min.	Max.		
1	TpC	OSCIN Clock Period	41.5		ns	1
			83		ns	2
2	TrC, TfC	OSCIN Rise and Fall Time		12	ns	
3	TwCL, TwCH	OSCIN Low and High Width	17	25	ns	1
			38		ns	2

Notes:
1. Clock divided by 2 internally (MODER.DIV2=1)
2. Clock not divided by 2 internally (MODER.DIV2=0)

CLOCK TIMING

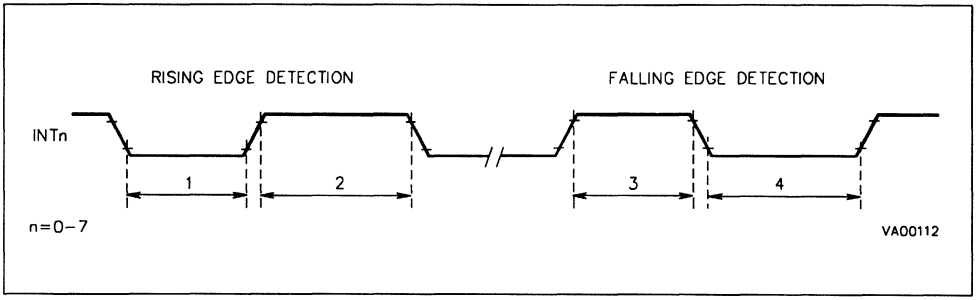


EXTERNAL INTERRUPT TIMING TABLE (V_{DD} = 5V ± 10%, T_A = 0°C to 70°C, Cload = 50pF, INTCLK = 12MHz, Push-pull output configuration, unless otherwise specified)

N°	Symbol	Parameter	Value (Note)				Unit
			OSCIN Divided By 2 Min.	OSCIN Not Divided By 2 Min.	Min.	Max.	
1	TwLR	Low Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95		ns
2	TwHR	High Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95		ns
3	TwHF	High Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95		ns
4	TwLF	Low Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95		ns

Note: The value left hand two columns show the formula used to calculate the timing minimum or maximum from the oscillator clock period, prescale value and number of wait cycles inserted.
The value right hand two columns show the timing minimum and maximum for an external clock at 24 MHz divided by 2, prescale value of zero and zero wait status.

EXTERNAL INTERRUPT TIMING

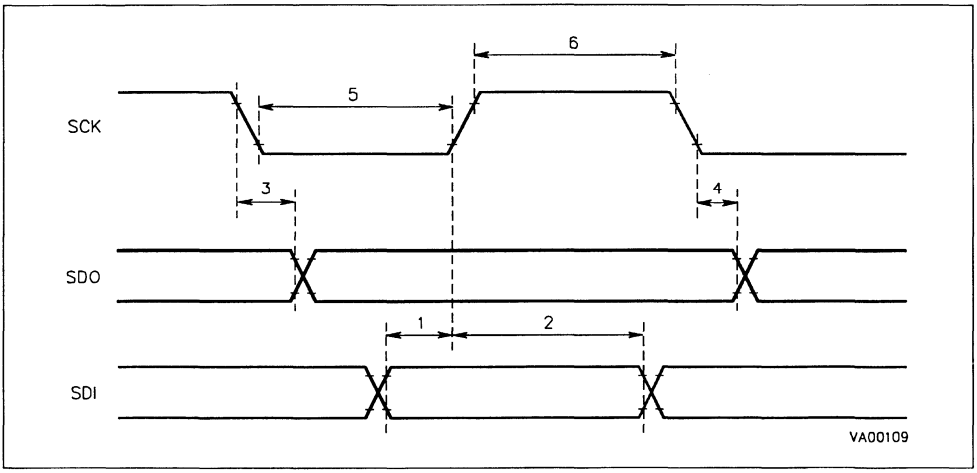


SPI TIMING TABLE (V_{DD} = 5V ± 10%, T_A = 0°C to 70°C, Cload = 50pF, INTCLK = 12MHz, Output Alternate Function set as Push-pull)

N°	Symbol	Parameter	Value		Unit
			Min.	Max.	
1	TsDI	Input Data Set-up Time	100		ns
2	ThDI (1)	Input Data Hold Time	1/2 TpC+100		ns
3	TdOV	SCK to Output Data Valid		100	ns
4	ThDO	Output Data Hold Time	-20		ns
5	TwSKL	SCK Low Pulse Width	300		ns
6	TwSKH	SCK High Pulse Width	300		ns

Note: TpC is the OSCIN Clock period.

SPI TIMING



ANALOG SPECIFICATIONS (V_{DD} = 5V ± 10%, T_A = +25°C)

Parameter	Min.	Typical	Max.	Units [1]	Notes
An. Input Range			AV_{CC}	V	
AV_{CC}	3		V_{CC}	V	
Conversion Time	4.5-5.5			µs	(2,3)
Sample Time	1.5			µs	(2)
Power-up Time	60			µs	
Resolution	6-8	6-8		bit	
Monotonicity	GUARANTEED				
No missing Codes					
Zero Input reading	00				
Full scale reading			FC-FF	LSBs	
Offset Error	TBD			LSBs	
Gain Error	-1		+1	LSBs	
Diff. Non lin.	-1		+1	LSBs	
Int Non Lin.	-1		+1	LSBs	
Absolute Accuracy	TBD			LSBs	
S/N	TBD			dB	
Input Resistance	8	12	15	KΩ	(4)
Hold capacitance Max.			4-15	pF	(5)
Input Leakage			3	mA	

Notes:
1. "LSBs", as used here, has a value of AV_{CC}/64 or /256
2. At 24MHz external clock
3. Including sample time
4. It must be intended as the internal series resistance before the sampling capacitor
5. For 6 or 8 bit converter respectively.

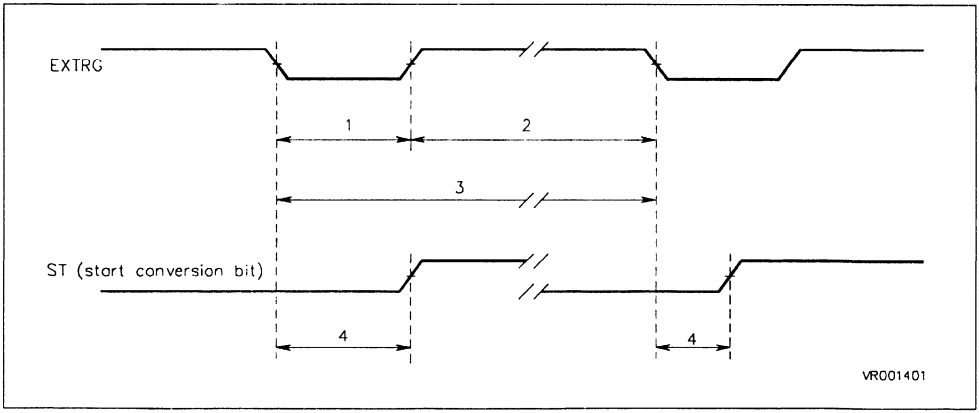
A/D CONVERTER

EXTERNAL TRIGGER TIMING (V_{DD} = 5V ± 10%, T_A = - 0°C to + 70°C, Cload = 50pF)

N°	Symbol	Parameter	Oscin divided by 2 (1)		Oscin not divided (1)		Value(2)		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
1	T_{LOW}	External Trigger pulse width	$2xT_{PC}$		T_{PC}		83		ns
2	T_{HIGH}	External Trigger pulse distance	$2xT_{PC}$		T_{PC}		83		ns
3	T_{EXT}	Eternal trigger active edges distance (fast mode)	$141xT_{PC}$		$70.5xT_{PC}$		5.87		µs
4	T_{STR}	Internal delay between EXTRG falling edge and first conversion start	T_{PC}	$3xT_{PC}$	$0.5xT_{PC}$	$1.5xT_{PC}$	41.5	125	ns

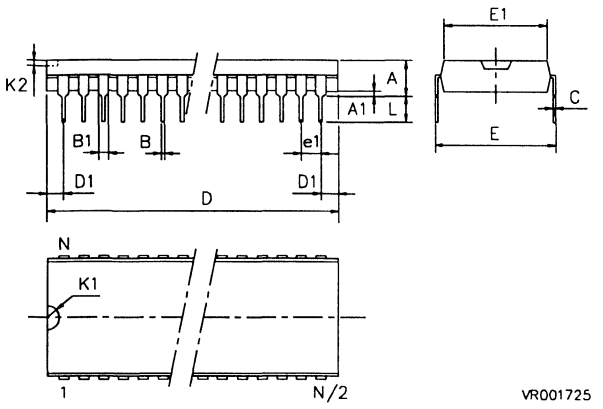
Notes:
1. Variable clock (TPC=OSCIN clock period)
2. CPUCLK=12MHz

A/D External Trigger Timing



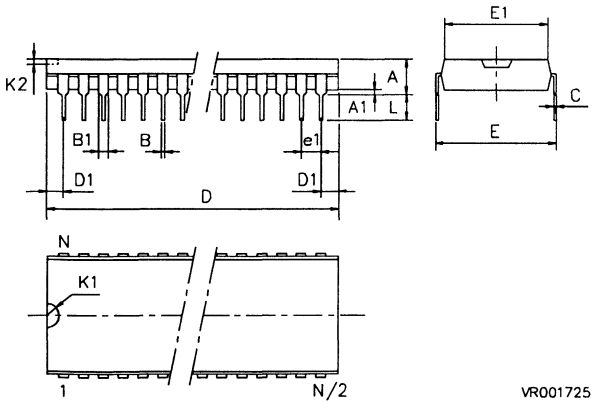
PACKAGE MECHANICAL DATA

42-Pin Plastic Shrink Dual-In-line Package, 600 Mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D	36.32		39.12	1.430		1.540
D1	–	–	–	–	–	–
E			18.54			0.730
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	42					

56-Pin Plastic Shrink Dual-In-line Package, 600 Mil Width



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D		52.12			2.052	
D1	–	–	–	–	–	–
E			18.54			0.730
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	56					

ORDERING INFORMATION

Sales Type	ROM Size	RAM Size	Temperature Range	Package
ST9294J6B1/XX	32K	640	0°C to+ 70°C	PSDIP42
ST9294J5B1/XX	24K	640		
ST9294J4B1/XX	24K	384		
ST9294J3B1/XX	16K	640		
ST9294J2B1/XX	16K	384		
ST9294J1B1/XX	12K	640		
ST9294N6B1/XX	32K	640	0°C to+ 70°C	PSDIP56
ST9294N5B1/XX	24K	640		
ST9294N4B1/XX	24K	384		
ST9294N3B1/XX	16K	640		
ST9294N2B1/XX	16K	384		
ST9294N1B1/XX	12K	640		

Note: "XX" is the ROM code identifier allocated by SGS-THOMSON after receipt of the related ROM file.

ST9294 OPTION LIST

Please copy this page (enlarge if possible) and complete ALL sections. Send the form, with the ROM code image required, to your local SGS-THOMSON sales office.

Customer Company : [.....]
Company Address : [.....]
[.....]

Telephone : [.....]
FAX : [.....]
Contact : [.....] Telephone (Direct) : [.....]

Please confirm device required :

SDIP42 Package

[ ] ST9294J1 (12K ROM, 640 RAM)
[ ] ST9294J2 (16K ROM, 384 RAM)
[ ] ST9294J3 (16K ROM, 640 RAM)
[ ] ST9294J4 (24K ROM, 384 RAM)
[ ] ST9294J5 (24K ROM, 640 RAM)
[ ] ST9294J6 (32K ROM, 640 RAM)

SDIP56 Package

[ ] ST9294N1 (12K ROM, 640 RAM)
[ ] ST9294N2 (16K ROM, 384 RAM)
[ ] ST9294N3 (16K ROM, 640 RAM)
[ ] ST9294N4 (24K ROM, 384 RAM)
[ ] ST9294N5 (24K ROM, 640 RAM)
[ ] ST9294N6 (32K ROM, 640 RAM)

Temperature Range : 0 to 70°C

Special Marking [ ] No
[ ] Yes 14 characters [| | | | | | | | | | | | | |] for SDIP42
16 characters [| | | | | | | | | | | | | | | |] for SDIP56
Authorized characters are letters, digits, '.', '-', '/' and spaces

Please consult your local SGS-THOMSON sales office for other marking details

OSD Code (INTEL ® HEX format files on PCcompatible disk)
[ ] Separate EVEN/ODD byte Filename [..... .EV%]
Filename [..... .OD%]
[ ] Single interleaved file Filename [.....]

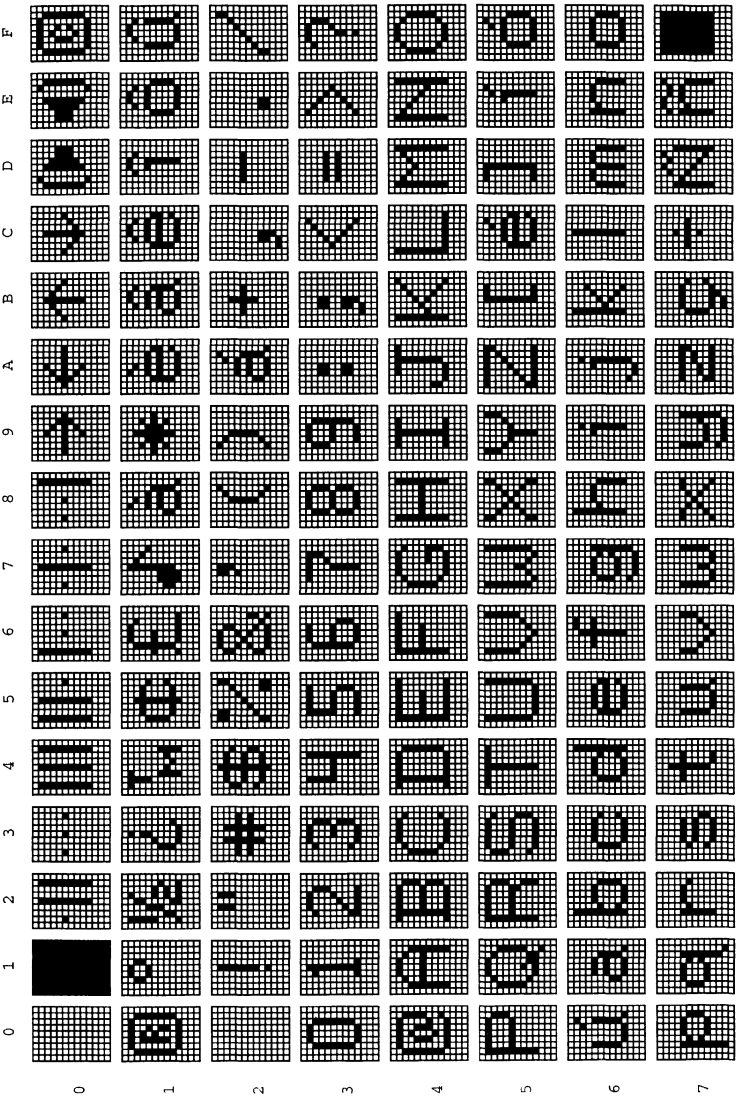
Code : [ ] ROM (27128, 27256)
[ ] INTEL ® HEX format file on PC disk Filename [.....]
Confirmation : [ ] Code checked with EPROM device in application

Yearly Quantity forecast : [.....] k units
for a period of : [.....] years
Preferred Production start dates : [.....] (YY/MM/DD)

Customer Signature :
Date :

APPENDIX 1

Sample OSD Character Set



VR0A1824

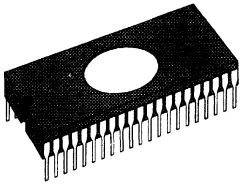
SGS-THOMSON MICROELECTRONICS

ST92E94
ST92T94

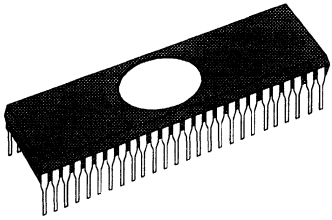
24K EPROM HCMOS MCUs WITH ON SCREEN DISPLAY AND CLOSED-CAPTION DATA SLICER

PRELIMINARY DATA

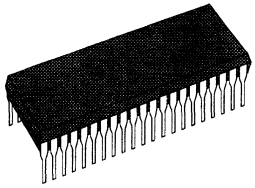
- Register oriented 8/16 bit CORE with RUN, WFI and HALT modes
- Minimum instruction cycle time: 500ns (12MHz internal)
- 24K bytes of EPROM, 640 bytes of RAM, 224 general purpose registers available as RAM, accumulators or index registers (Register File)
- 42-lead or 56-lead Window Ceramic Shrink DIP package for ST92E94
- 42-lead or 56-lead Plastic Shrink DIP package for ST92T94
- Interrupt handler and Serial Peripheral Interface as standard features
- 31 (42 pin package) / 42 (56 pin package) fully programmable I/O pins
- 34 character x15 rows software programmable On Screen Display module with colour, italic, underline, flash, transparent and fringe attribute options
- Digital Data Slicer extracting closed caption data from video
- 8 8-bit PWM D/A outputs with repetition frequency 2 to 32kHz and 12V Open Drain Capability
- 16 bit Timer with 8 bit Prescaler, able to be used as a Watchdog Timer
- 16-bit programmable Slice Timer with 8-bit prescaler
- 3 channel Analog to Digital Converter, with integral sample and hold, fast 5.75µs conversion time, 6-bit guaranteed resolution
- Rich Instruction Set and 14 Addressing modes
- Division-by-Zero trap generation
- Versatile Development tools, including assembler, linker, C-compiler, archiver, graphic oriented debugger and hardware emulators
- Real Time Operating System
- Compatible with ST9294 ROM device



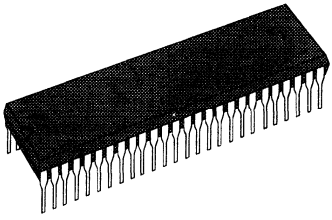
CSDIP42-W



CSDIP56-W



PSDIP42



PSDIP56

(Ordering Information at the end of the datasheet)

Figure 1a. 42 Pin Shrink DIP Pinout

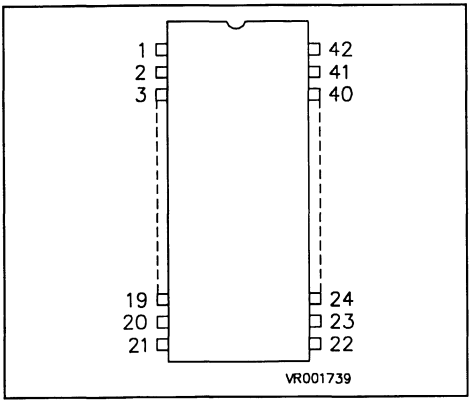
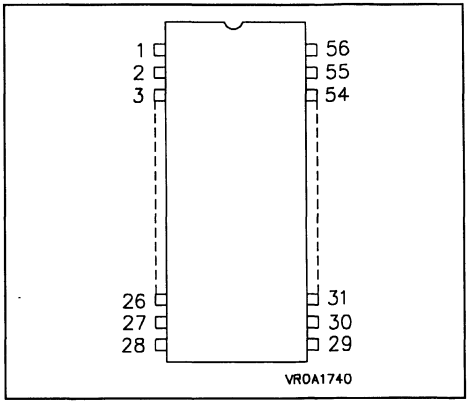


Figure 1b. 56 Pin Shrink DIP Pinout



ST92E94J Pin Description

Pin	Pin name	Pin	Pin name
1	P2.0/INT7	42	P2.1/INT5/AIN1
2	$\overline{\text{RESET}}/V_{PP}$	41	P2.2/INT0/AIN2
3	P0.7	40	P2.3/INT6/SDO
4	P0.6	39	P2.4/NMI/$\overline{\text{PD}}$
5	P0.5	38	P2.5/AIN3
6	P0.4	37	OSCIN
7	P0.3	36	OSCOUT
8	P0.2	35	P4.7/PWM7/ EXTRG (AD)
9	P0.1	34	P4.6/PWM6
10	P0.0	33	P4.5/PWM5
11	CCVideo	32	P4.4/PWM4
12	P3.6	31	P4.3/PWM3
13	P3.5	30	P4.2/PWM2
14	P3.4	29	P4.1/PWM1
15	P3.3/B	28	P4.0/PWM0
16	P3.2/G	27	VSYNC
17	P3.1/R	26	HSYNC
18	P3.0/FB	25	AV_{DD}
19	P5.1/SDIO	24	PLLR
20	P5.0/SCK/INT2	23	PLLF
21	V_{DD}	22	V_{SS}

ST92E94N Pin Description

Pin	Pin name	Pin	Pin name
1	P2.1/INT5/AIN1	56	P2.2/INT0/AIN2
2	P2.0/INT7	55	P2.3/INT6/SDO
3	$\overline{\text{RESET}}/V_{PP}$	54	P2.4/NMI
4	P0.7	53	P2.5/AIN3
5	P0.6	52	P1.0
6	P0.5	51	P1.1
7	NC	50	P1.2
8	P0.4	49	P1.3
9	P0.3	48	P1.4
10	P0.2	47	P1.5
11	P0.1	46	P1.6
12	P0.0	45	P1.7
13	NC	44	OSCIN
14	V_{DD}	43	OSCOUT
15	CCVideo	42	P4.7/PWM7/ EXTRG (AD)
16	P3.7	41	P4.6/PWM6
17	P3.6	40	P4.5/PWM5
18	P3.5	39	P4.4/PWM4
19	P3.4	38	P4.3/PWM3
20	P3.3/B	37	P4.2/PWM2
21	P3.2/G	36	P4.1/PWM1
22	P3.1/R	35	P4.0/PWM0
23	P3.0/FB	34	VSYNC
24	P5.3	33	HSYNC
25	P5.2	32	AV_{DD}
26	P5.1/SDIO	31	PLLR
27	P5.0/SCK/INT2	30	PLLF
28	V_{DD}	29	V_{SS}

16.1 GENERAL DESCRIPTION

The ST92E94 are EPROM members in windowed ceramic (E) and plastic OTP (T) packages, completely of the ST9 family of microcontrollers, completely developed and produced by SGS-THOMSON Microelectronics using a n-well proprietary HCMOS process.

The EPROM parts are fully compatible with their ROM versions and this datasheet will thus provide only information specific to the EPROM based devices.

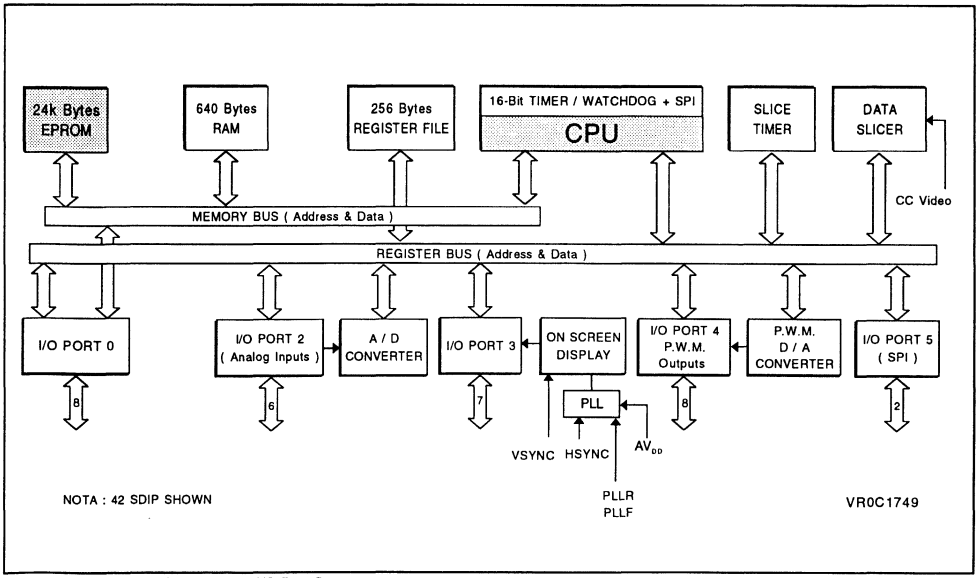
THE READER IS ASKED TO REFER TO THE DATASHEET OF THE ST9294 ROM-BASED DEVICE FOR FURTHER DETAILS.

The EPROM ST92E94 may be used for the prototyping and pre-production phases of development.

The nucleus of the ST92E94 is the advanced Core which includes the Central Processing Unit (CPU), the Register File, a 16-bit Timer/Watchdog with 8-bit Prescaler, a Serial Peripheral Interface supporting S-bus, I^2C-bus and IM-bus Interface, plus two 8-bit I/O ports. The Core has independent memory and register buses allowing a high degree of pipelining to add to the efficiency of the code execution speed of the extensive instruction set.

The powerful I/O capabilities demanded by microcontroller applications are fulfilled by the ST92E94 with up to 31/42 I/O lines dedicated to digital Input/Output. These lines are grouped into up to six I/O Ports and can be configured on a bit basis under software control to provide timing, status signals, timer inputs and outputs, analog inputs, external interrupts and serial or parallel I/O.

Figure 2.ST92E94 Block Diagram



Note : Refer to Table 1 for ST92E94 I/O Port Summary

GENERAL DESCRIPTION (Continued)

Three basic memory spaces are available to support this wide range of configurations: Program Memory (internal), Data Memory (intenal) and the Register File, which includes the control and status registers of the on-chip peripherals.

The 16-bit Slice Timer with an 8-bit Prescaler.

The human interface is provided by the On Screen Display module, this can produce up to 15 lines of up to 34 characters from a ROM defined 128 character set. The 9x13 character can be modified by 4 different pixel sizes, with character rounding, and formed into words with colour and format attributes.

Closed Caption control for the display of information transmitted through the video input is facilitated with the Data Slicer. This module has manual and automatic Slicing levels for both Sync and Data and allows the user to select the video line containing the data relative to the Vertical synchronisation pulse.

Control of TV settings is able to be made with up to eight 8-bit PWM outputs, with a frequency maximum of 23,437Hz at 8-bit resolution (INTCLK = 12MHz). Low resolutions with higher frequency operation can be programmed.

In addition there is a 3 channel Analog to Digital Converter with integral sample and hold, fast 5.75µs conversion time and 6-bit guaranteed resolution.

16.2 PIN DESCRIPTION

VSYNC. *Vertical Synch.* Vertical video synchronisation input to OSD. Positive or negative polarity.

HSYNC. *Horizontal Synch.* Horizontal video synchronisation input to OSD. Positive or negative polarity.

CCVideo. *Composite Video Input.* Input to Data Slicer for Closed Caption extraction, 1V± 6dB or 2V± 3dB.

PLLF. *PLL Filter input.* Filter input for the OSD for PLL feed-back.

PLLR. *PLL Resistor connection pin.* For resistor connection to select the PLL gain adjust.

$\overline{\text{RESET}}$/V_{PP}. *Reset (input, active low) or V_{PP} (input).* The ST9 is initialised by the Reset signal. With the deactivation of $\overline{\text{RESET}}$, program execution begins from the Program memory location pointed to by the vector contained in program memory locations 00h and 01h. In the EPROM programming Mode, this pin acts as the programming voltage input V_{PP}.

OSCIN, OSCOUT. *Oscillator (input and output).* These pins connect a parallel-resonant crystal (24MHz maximum), or an external source to the on-chip clock oscillator and buffer. OSCin is the input of the oscillator inverter and internal clock generator; OSCout is the output of the oscillator inverter.

AV_{DD}. Analog V_{DD} of PLL. This pin must be tied to V_{DD} externally to the ST92E94.

V_{DD}. Main Power Supply Voltage (5V±10%)

V_{SS}. Digital Circuit Ground.

P0.0-P0.7, P2.0-P2.5, P3.0-P3.6, P4.0-P4.7, P5.0-P5.1 (J suffix)
P0.0-P0.7, P1.0-P1.7, P2.0-P2.5, P3.0-P3.7, P4.0-P4.7, P5.0-P5.3 (N suffix) *I/O Port Lines (Input/Output, TTL or CMOS compatible).* 31/42 lines grouped into I/O ports, bit programmable under program control as general purpose I/O or as Alternate functions (see next section).
P4.0-P4.7 are high voltage (12V) open drain

I/O Port Alternate Functions.

Each pin of the I/O ports of the ST92E94 may assume software programmable Alternative Functions as shown in the Pin Configuration Drawings. Table 1 shows the Functions allocated to each I/O Port pin.

PIN DESCRIPTION (Continued)

Table 1. ST92E94 I/O Port Alternative Function Summary

I/O PORT	Name	Function	Alternate Function	Pin Assignment	
Port.bit				92E94J	92E94N
P0.0		I/O		10	12
P0.1		I/O		9	11
P0.2		I/O		8	10
P0.3		I/O		7	9
P0.4		I/O		6	8
P0.5		I/O		5	6
P0.6		I/O		4	5
P0.7		I/O		3	4
P1.0		I/O		-	52
P1.1		I/O		-	51
P1.2		I/O		-	50
P1.3		I/O		-	49
P1.4		I/O		-	48
P1.5		I/O		-	47
P1.6		I/O		-	46
P1.7		I/O		-	45
P2.0	INT7	I	External Interrupt 7 with Schmitt Trigger	1	2
P2.1	INT5	I	External Interrupt 5 with Schmitt Trigger	42	1
P2.1	AIN1	I	A/D Analog Input 1	42	1
P2.2	INT0	I	External Interrupt 0	41	56
P2.2	AIN2	I	A/D Analog Input 2	41	56
P2.3	INT6	I	External Interrupt 6	40	55
P2.3	SDO	O	MSPI Serial Data Output	40	55
P2.4	NMI	I	Non-Maskable Interrupt	39	54
P2.5	AIN3	I	A/D Analog Input 3	38	53
P3.0	FB	O	Fast Blanking OSD output	18	23
P3.1	R	O	Red Video Colour OSD output	17	22
P3.2	G	O	Green Video Colour OSD output	16	21
P3.3	B	O	Blue Video Colour OSD output	15	20

PIN DESCRIPTION (Continued)

Table 1. ST92E94 I/O Port Alternative Function Summary(Continued)

I/O PORT	Name	Function	Alternate Function	Pin Assignment	
Port.bit				92E94J	92E94N
P3.4		I/O		14	19
P3.5		I/O		13	18
P3.6		I/O		12	17
P3.7		I/O		-	16
P4.0	PWM0	O	PWM Output 0	28	35
P4.1	PWM1	O	PWM Output 1	29	36
P4.2	PWM2	O	PWM Output 2	30	37
P4.3	PWM3	O	PWM Output 3	31	38
P4.4	PWM4	O	PWM Output 4	32	39
P4.5	PWM5	O	PWM Output 5	33	40
P4.6	PWM6	O	PWM Output 6	34	41
P4.7	PWM7	O	PWM Output 7	35	42
P4.7	EXTRG	I	A/D External Trigger	35	42
P5.0	SCK	O	SPI Serial Clock (1)	20	27
P5.0	INT2	I	External Interrupt 2 (1)	20	27
P5.1	SDIO	O	SPI Serial Data Input/Output (1)	19	26
P5.2		I/O		-	25
P5.3		I/O		-	24

Notes.
1. The alternate functions of SCK/INT2 and SDIO may be swapped by using the SWAP Register Function.
2. Schmitt trigger options are available as a mask option for any input pin.

16.3 MEMORY

The memory of the ST92E94 is functionally divided into two areas, the Register File and Memory. The Memory may optionally be divided into two spaces, each having a maximum of 65,536 bytes. The two memory spaces are separated by function, one space for Program code, the other for Data. The ST92E94 24K bytes of on-chip ROM memory is selected at memory addresses 0 through 5FFFh (hexadecimal) in the PROGRAM space while the ST92T94 OTP version has the top 64 bytes of the EPROM reserved by SGS-THOMSON for testing purposes (5FC0h - 5FFFh). The 640 bytes of on-chip RAM are mapped into both Program and Data space (the P/D bit is not used for the decoding) at addresses 6000h through 627Fh.

The on-chip general purpose (GP) Registers may also be used as RAM memory for minimum chip count systems.

16.4 EPROM PROGRAMMING

The 24576 bytes of EPROM memory of the ST92E94 (24512 for the ST92T94) may be programmed by using the EPROM Programming Boards (EPB) available from SGS-THOMSON.

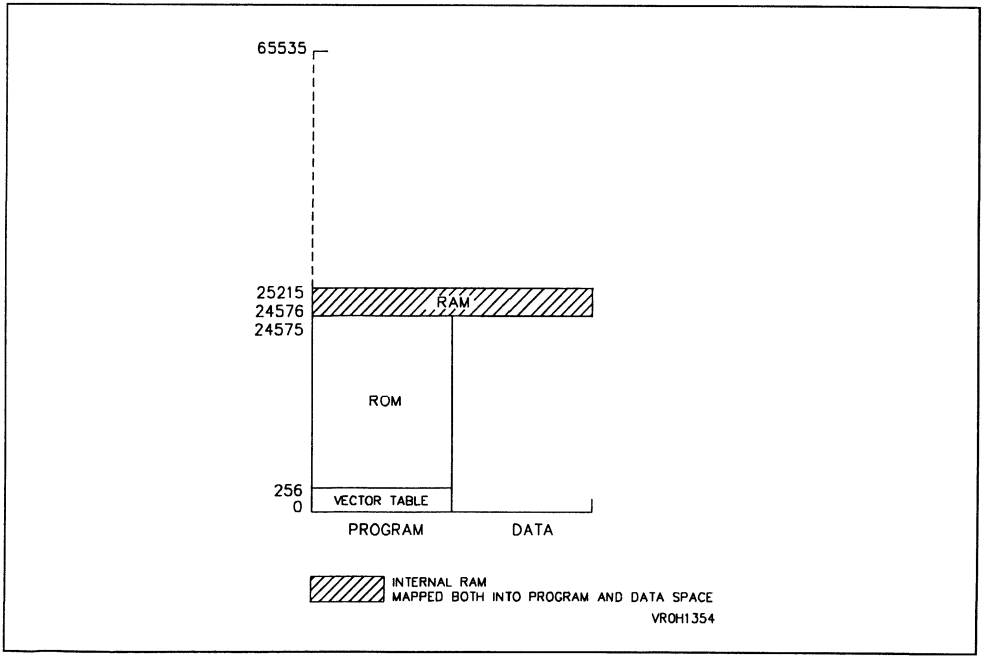
16.4.1 Eprom Erasing

The EPROM of the windowed package of the ST92E94 may be erased by exposure to Ultra-Violet light.

The erasure characteristic of the ST92E94 is such that erasure begins when the memory is exposed to light with a wave lengths shorter than approximately 4000Å. It should be noted that sunlight and some types of fluorescent lamps have wavelengths in the range 3000-4000Å. It is thus recommended that the window of the ST92E94 packages be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.

The recommended erasure procedure of the EPROM is the exposure to short wave ultraviolet light which have a wave-length 2537Å. The integrated dose (i.e. U.V. intensity x exposure time) for erasure should be a minimum of 15W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm^2 power rating. The ST92E94 should be placed within 2.5cm (1Inch) of the lamp tubes during erasure.

Figure 3. Memory Map



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{DD}	Supply Voltage	V_{SS} – 0.3 to 7.0	V
AV_{DD}	Analog Supply Voltage	AVDD ≥ V_{SS} - 0.3 to 7.0 V_{DD} - 0.3 ≤ AV_{DD} ≤ V_{DD} + 0.3	V
V_I	Input Voltage	V_{SS} – 0.3 to V_{DD} + 0.3	V
V_O	Output Voltage	V_{SS} – 0.3 to V_{DD} + 0.3	V
T_{STG}	Storage Temperature	– 55 to + 150	°C

Note: Stresses above those listed as "absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Value		Unit
		Min.	Max.	
T_A	Operating Temperature	0	70	°C
V_{DD}	Operating Supply Voltage	4.5	5.5	V
f_{OSCE}	External Oscillator Frequency		24	MHz
f_{OSCI}	Internal Oscillator Frequency		12	MHz

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5V ± 10% T_A = 0°C to + 70°C, unless otherwise specified)

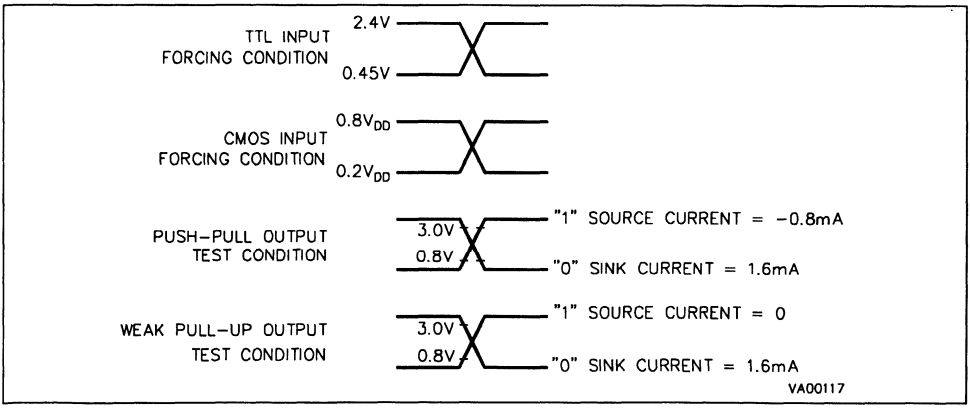
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{IHCK}	Clock Input High Level	External Clock	0.7 V_{DD}			V
V_{ILCK}	Clock Input Low Level	External Clock			0.3 V_{DD}	V
V_{IH}	Input High Level	TTL	2.0			V
		CMOS	0.7 V_{DD}			V
V_{IL}	Input Low Level	TTL			0.8	V
		CMOS			0.3 V_{DD}	V
V_{IHRS}	$\overline{\text{RESET}}$ Input High Level		0.7 V_{DD}			V
V_{ILRS}	$\overline{\text{RESET}}$ Input Low Level				0.3 V_{DD}	V
V_{HYRS}	$\overline{\text{RESET}}$ Input Hysteresis		0.3			V
V_{IHY}	P2.0, P2.1 Input Hysteresis		0.9		1.5	V
V_{HYHV}	HSYNC/VSYNC Hysteresis		0.5			V

ELECTRICAL CHARACTERISTICS (Continued)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
V_{OH}	Output High Level	Push Pull, Iload = 4mA	VDD – 0.8			V
V_{OL}	Output Low Level	Push Pull or Open Drain, Iload = 2mA			0.4	V
I_{WPU}	Weak Pull-up Current	Bidirectional Weak Pull-up, V_{OL} = 0V	– 80	– 200	– 420	µA
I_{OH}	High Voltage Open Drain Output Leakage Current	V_{OH} =12V			+ 30	
I_{LKIO}	I/O Pin Input Leakage	Input/Tri-State, 0V < VIN < V_{DD}	– 10		+ 10	µA
I_{LKRS}	$\overline{\text{RESET}}$ Pin Input Leakage	0V < V_{IN} < V_{DD}	– 30		+ 30	µA
I_{LKAD}	A/D Pin Input Leakage	Alternate Function, Open Drain, 0V < V_{IN} < V_{DD}	– 3		+ 3	µA
I_{LKOS}	OSCIN Pin Input Leakage	0V < V_{IN} < V_{DD}	– 10		+ 10	µA

Note:
1. All I/O Ports are configured in Bidirectional Weak Pull-up Mode with no DC load, External Clock pin (OSCIN) is driven by square wave external clock. No peripheral working (Internal Program Execution).

AC TEST CONDITIONS



AC ELECTRICAL CHARACTERISTICS

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
I_{DD}	Run Mode Current no CPUCLK prescale, CLOCK divide by 2	24MHz, Note 1		40	70	mA
I_{DP2}	Run Mode Current Prescale by 2 CLOCK divide by 2	24MHz, Note 1		19	40	mA
I_{WFI}	WFI Mode Current no CPUCLK prescale, CLOCK divide by 2	24MHz, Note 1		15	20	mA
I_{HALT}	HALT Mode Current	24MHz, Note 1		50	100	µA
I_{LPR}	Low Power Reset Current, Note 2			TBD	TBD	µA

Notes:
1. All I/O Ports are configured in Bidirectional Weak Pull-up Mode with no DC load, External Clock pin (OSCIN) is driven by square wave external clock. No peripheral working.
2. All I/O Port connected to V_{DD} by 10kΩPull-up, External Clock pin (OSCIN) Stopped.

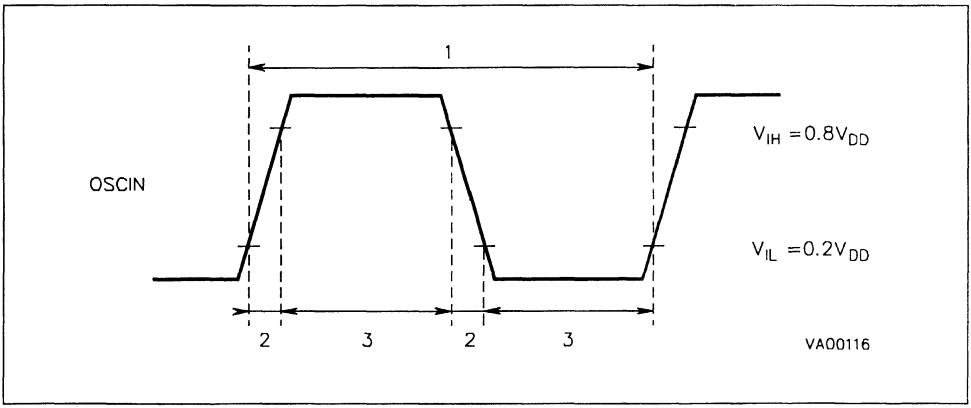
CLOCK TIMING TABLE

(V_{DD} = 5V ± 10%, T_A = − 40°C to + 85°C, INTCLK = 12MHz, unless otherwise specified)

N°	Symbol	Parameter	Value		Unit	Note
			Min.	Max.		
1	TpC	OSCIN Clock Period	41.5		ns	1
			83		ns	2
2	TrC, TfC	OSCIN Rise and Fall Time		12	ns	
3	TwCL, TwCH	OSCIN Low and High Width	17	25	ns	1
			38		ns	2

Notes:
1. Clock divided by 2 internally (MODER.DIV2=1)
2. Clock not divided by 2 internally (MODER.DIV2=0)

CLOCK TIMING

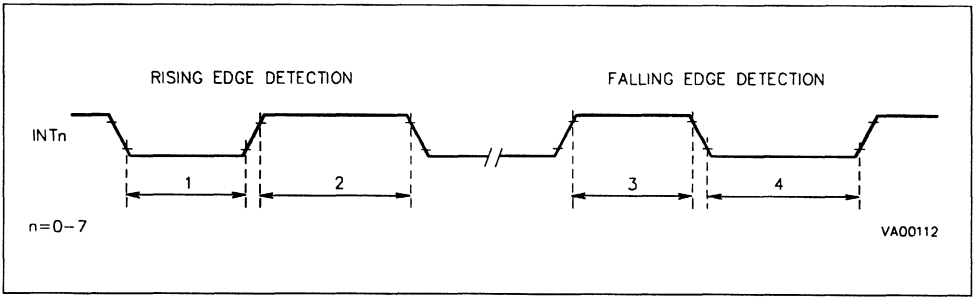


EXTERNAL INTERRUPT TIMING TABLE (V_{DD} = 5V ± 10%, T_A = 0°C to 70°C, Cload = 50pF, INTCLK = 12MHz, Push-pull output configuration, unless otherwise specified)

N°	Symbol	Parameter	Value (Note)				Unit
			OSCIN Divided By 2 Min.	OSCIN Not Divided By 2 Min.	Min.	Max.	
1	TwLR	Low Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95		ns
2	TwHR	High Level Minimum Pulse Width in Rising Edge Mode	2TpC+12	TpC+12	95		ns
3	TwHF	High Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95		ns
4	TwLF	Low Level Minimum Pulse Width in Falling Edge Mode	2TpC+12	TpC+12	95		ns

Note: The value left hand two columns show the formula used to calculate the timing minimum or maximum from the oscillator clock period, prescale value and number of wait cycles inserted.
The value right hand two columns show the timing minimum and maximum for an external clock at 24 MHz divided by 2, prescale value of zero and zero wait status.

EXTERNAL INTERRUPT TIMING

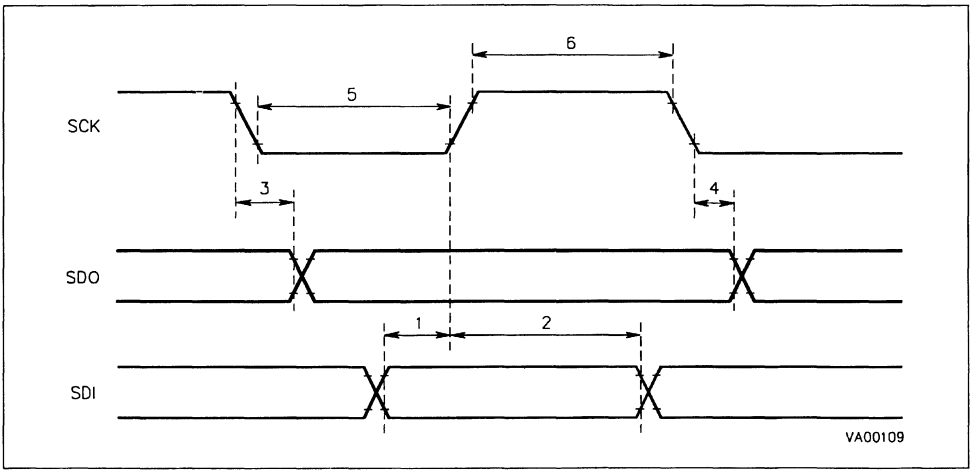


SPI TIMING TABLE (V_{DD} = 5V ± 10%, T_A = 0°C to 70°C, Cload = 50pF, INTCLK = 12MHz, Output Alternate Function set as Push-pull)

N°	Symbol	Parameter	Value		Unit
			Min.	Max.	
1	TsDI	Input Data Set-up Time	100		ns
2	ThDI (1)	Input Data Hold Time	1/2 TpC+100		ns
3	TdOV	SCK to Output Data Valid		100	ns
4	ThDO	Output Data Hold Time	-20		ns
5	TwSKL	SCK Low Pulse Width	300		ns
6	TwSKH	SCK High Pulse Width	300		ns

Note: TpC is the OSCIN Clock period.

SPI TIMING



ANALOG SPECIFICATIONS (V_{DD} = 5V ± 10%, T_A = +25°C)

Parameter	Min.	Typical	Max.	Units [1]	Notes
An. Input Range			AV_{CC}	V	
AV_{CC}	3		V_{CC}	V	
Conversion Time	4.5-5.5			µs	(2,3)
Sample Time	1.5			µs	(2)
Power-up Time	60			µs	
Resolution	6-8	6-8		bit	
Monotonicity	GUARANTEED				
No missing Codes					
Zero Input reading	00				
Full scale reading			FC-FF	LSBs	
Offset Error	TBD			LSBs	
Gain Error	-1		+1	LSBs	
Diff. Non lin.	-1		+1	LSBs	
Int Non Lin.	-1		+1	LSBs	
Absolute Accuracy	TBD			LSBs	
S/N	TBD			dB	
Input Resistance	8	12	15	KΩ	(4)
Hold capacitance Max.			4-15	pF	(5)
Input Leakage			3	mA	

Notes:
1. "LSBs", as used here, has a value of AV_{CC}/64 or /256
2. At 24MHz external clock
3. Including sample time
4. The internal series resistance before the sampling capacitor
5. For 6 or 8 bit converter respectively.

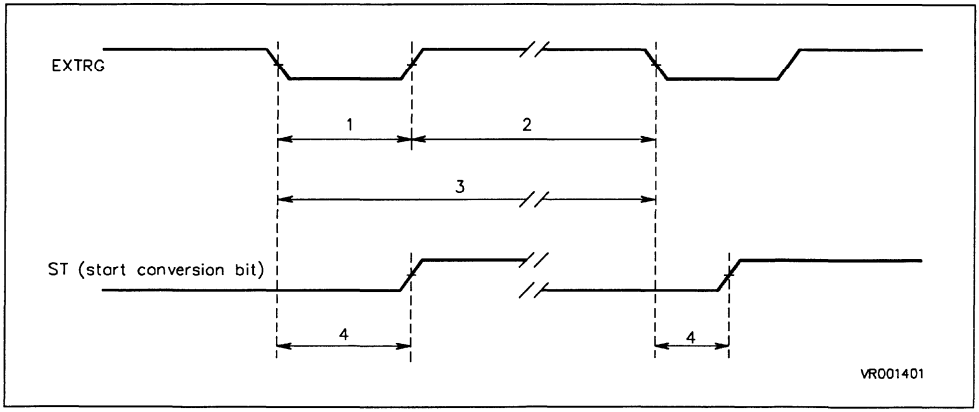
A/D CONVERTER

EXTERNAL TRIGGER TIMING (V_{DD} = 5V ± 10%, T_A = 0°C to + 70°C, Cload = 50pF)

N°	Symbol	Parameter	Oscin divided by 2 (1)		Oscin not divided (1)		Value(2)		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
1	T_{LOW}	External Trigger pulse width	$2xT_{PC}$		T_{PC}		83		ns
2	T_{HIGH}	External Trigger pulse distance	$2xT_{PC}$		TPC		83		ns
3	T_{EXT}	External trigger active edges distance (fast mode)	$141xT_{PC}$		70.5xTPC		5.87		µs
4	T_{STR}	Internal delay between EXTRG falling edge and first conversion start	T_{PC}	3xTPC	$0.5xT_{PC}$	$1.5xT_{PC}$	41.5	125	ns

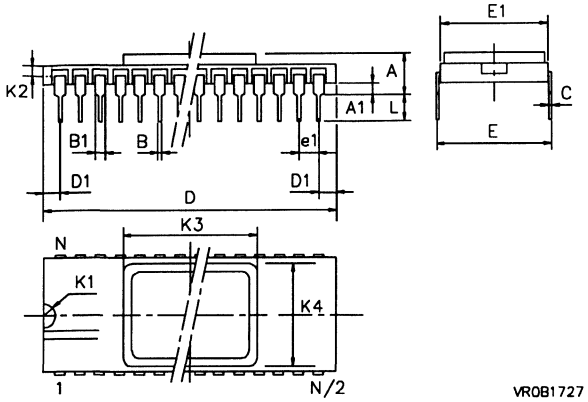
Notes:
1. Variable clock (TPC=OSCIN clock period)
2. CPUCLK=12MHz

A/D External Trigger Timing



PACKAGE MECHANICAL DATA

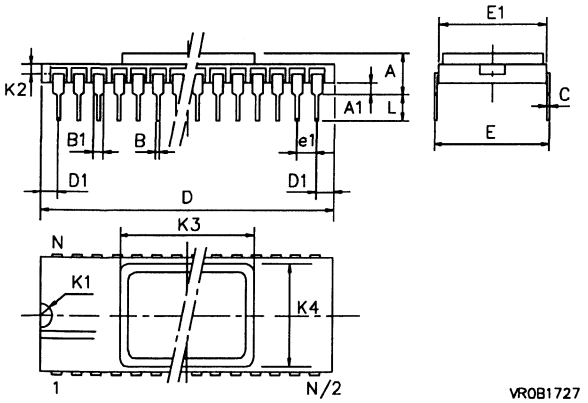
42-Pin Ceramic Shrink Dual-In-line Package, 600 Mil Width with Window



VR0B1727

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D	36.32		39.12	1.430		1.540
D1	–	–	–	–	–	–
E	15.2 BSD			0.600 BSD		
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	42					

56-Pin Ceramic Shrink Dual-In-line Package, 600 Mil Width with Window



VR0B1727

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			0.02			0.200
A1	0.51			0.020		
B	0.35		0.59	0.014		0.023
B1	0.75		1.42	0.030		0.056
C	0.20		0.36	0.008		0.014
D		50.8			2.00	
D1	–	–	–	–	–	–
E			18.54			0.730
E1		13.72				0.540
K1	–	–	–	–	–	–
K2	–	–	–	–	–	–
K3		18.94			0.746	
K4		14.37			0.566	
L	2.54		3.81	.100		0.150
e1		1.78			0.070	
Number of Pins						
N	56					

ST92E94 STANDARD OPTION LIST
Please copy this page (enlarge if possible) and complete ALL sections.
Send the form, with the ROM code image required, to your local SGS-THOMSON sales office.

Customer Company : [.....]
Company Address : [.....]
[.....]
[.....]

Telephone : [.....]
FAX : [.....]
Contact : [.....] Telephone (Direct) : [.....]

Please confirm characteristics of device :

Device / Package

[ ] ST92E94J5 (24K ROM, 640 RAM) CSDIP42-W
[ ] ST92T94J5 (24K ROM, 640 RAM) PSDIP42
[ ] ST92E94N5 (24K ROM, 640 RAM) CSDIP56-W
[ ] ST92T94N5 (24K ROM, 640 RAM) PSDIP56

Temperature Range : 0 to +70°C

Notes :

OSD Code (INTEL ® HEX format files on PCcompatible disk)
[ ] Separate EVEN/ODD byte Filename [..... .EV%]
Filename [..... .OD%]
[ ] Single interleaved file Filename [.....]

Yearly Quantity forecast : [.....] k units
for a period of : [.....] years
Preferred Production start dates [.....] (YY/MM/DD)

Customer Signature :
Date :

16.5 ORDERING INFORMATION

Sales Type	EPROM Size (bytes)	RAM Size (bytes)	Temperature Range	Package
ST92E94J5F1/XX	24K	640	0°C to+ 70°C	CSDIP42-W
ST92E94N5F1/XX	24K	640		CSDIP56-W
ST92T94J5B1/XX	24K[1]	640		PSDIP42
ST92T94N5B1/XX	24K[1]	640		PSDIP56

Note: "XX" is the OSD ROM code identifier allocated by SGS-THOMSON after receipt of the related ROM file.
[1] The top 64 bytes of the EPROM Memory is reserved by SGS-THOMSON.

SGS-THOMSON MICROELECTRONICS

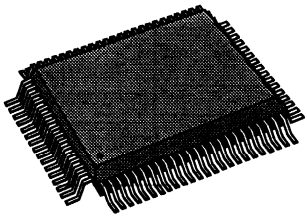
ST90R50
ST90R51

ROMLESS HCMOS MCU WITH BANKSWITCH AND A/D CONVERTER

PRELIMINARY DATA

- Register oriented 8/16 bit CORE with RUN, WFI and HALT modes
- Minimum instruction cycle time : 500ns (12MHz internal)
- 224 general purpose registers available as RAM, accumulators or index pointers (register file)
- ROMless to allow maximum external memory flexibility
- Bankswitch logic allowing a maximum addressing capability of 8Mbytes for Program and Dataspace (16Mbytes total)
- 84-pin plastic leadless chip carrier (ST90R50)
- 80-pin Plastic Quad Flat Pack package (ST90R51)
- DMA controller, Interrupt handler and a Serial Peripheral Interface as standard features
- 56 fully programmable I/O ports (ST90R50) 52 fully programmable I/O ports (ST90R51)
- Up to 8 external plus 1 non-maskable interrupts
- 16 bit Timer with 8 bit Prescaler, able to be used as a Watchdog Timer
- Three 16 bit Multifunction Timers, each with an 8 bit prescaler and 13 operating modes
- 8 channel 8 bit Analog to Digital Converter, with Analog Watchdogs and external references (ST90R50 only)
- Two Serial Communication Interfaces with asynchronous and synchronous capability
- Rich Instruction Set with 14 Addressing modes
- Division-by-Zero trap generation
- Versatile Development Tools, including assembler, linker, C-compiler, archiver, graphic oriented debugger and hardware emulators
- Real Time Operating System

PLCC84



PQFP80

(Ordering Information at the end of the Datasheet)

Table 1. ST90R50 Pin Description (PLCC84 Package)

Pin	Name
75	P86/INT7 /T3OUTA
76	P85/P/D/T3INB
77	P84/T1INA /WAIT/WDOUT
78	P83/T1OUTB /INT3/RDSTB5
79	P82/T1OUTA /INT1/WRSTB5
80	P81/T1INB /RDRDY5
81	P80/T3INA /WRRDY5
82	V_{DD}
83	P57/T3OUTB
84	P56/T3OUTA
● 1	P55/T1OUT1
2	P54/T1OUTA
3	P53/P/D
4	P52
5	P51
6	P50
7	OSCOUT
8	V_{SS}
9	OSCIN
10	RESET
11	P10/A8

Pin	Name
12	P11/A9
13	P12/A10
14	P13/A11
15	P14/A12
16	P15/A13
17	P16/A14
18	P17/A15
19	P60/A0
20	P61/A1
21	P62/A2
22	P63/A3
23	P64/A4
24	P65/A5
25	P66/A6
26	P67/A7
27	P00/AD0
28	P01/AD1
29	P02/AD2
30	P03/AD3
31	P04/AD4
32	P05/AD5

Pin	Name
53	P25/BS5
52	P24/BS4
51	P23/BS3
50	P22/BS2
49	P21/BS1
48	P20/BS0
47	P47/T0OUTA /BUSREQ
46	P46/INT5 /T0OUTB
45	P45/INT4 /BUSACK
44	P44/INT0 /WDOUT
43	P43
42	P42
41	P41
40	P40
39	R/W
38	DS
37	AS
36	V_{DD}
35	P07/AD7
34	V_{SS}
33	P06/AD6

Pin	Name
74	P87/NMI /T3OUTB
73	V_{SS}
72	AV_{DD}
71	P77/AIN7
70	P76/AIN6
69	P75/AIN5
68	P74/AIN4
67	P73/AIN3 /T0INA/P/D
66	P72/AIN2 /CLK0OUT /TX0CKIN /BSL_EN1
65	P71/AIN1 /T0INB/SDI /RD4RDY
64	P70/AIN0 /RX0CKIN /WDIN /BSH_EN1 /WR4RDY
63	P97/INT6/SDO /RD4STB
62	P96/INT2/SCK /WR4STB
61	P95/S0IN /BUSACK
60	P94/S0OUT
59	P93/T0OUTA /RX1CKIN
58	P92/TX1CKIN /CLK1OUT
57	P91/T0OUTB /S1IN
56	P90/S1OUT
55	P27/BS7
54	P26/BS6

Figure 1-1. 84 Pin PLCC Package

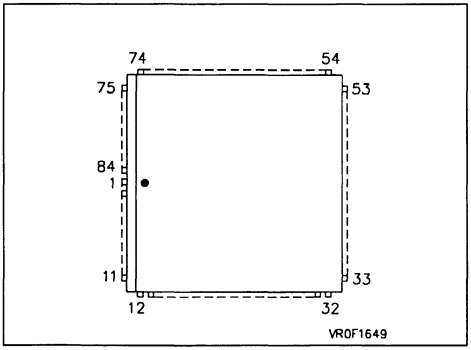


Table 2. ST90R50 Pin Description (PQFP80 Package)

Pin	Name
1	P10/A8
2	P11/A9
3	P12/A10
4	P13/A11
5	P14/A12
6	P15/A13
7	P16/A14
8	P17/A15
9	P60/A0
10	P61/A1
11	P62/A2
12	P63/A3
13	P64/A4
14	P65/A5
15	P66/A6
16	P67/A7
17	P00/A0/D0
18	P01/A1/D1
19	P02/A2/D2
20	P03/A3/D3
21	P04/A4/D4
22	P05/A5/D5
23	P06/A6/D6
24	V_{SS}

Pin	Name
25	P07/A7/D7
26	V_{DD}
27	$\overline{AS}$
28	$\overline{DS}$
29	R/$\overline{W}$
30	P40
31	P41
32	P42
33	P43
34	P44/INT0 /WDOUT
35	P45/INT4 /$\overline{BUSACK}$
36	P46/INT5 /T0OUTB
37	P47/T0OUTA /$\overline{BUSREQ}$
38	P20/BS0
39	P21/BS1
40	P22/BS2

Pin	Name
64	P87/NMI /T3OUTB
63	AV_{SS}
62	AV_{DD}
61	P77/AIN7
60	P76/AIN6
59	P75/AIN5
58	P74/AIN4
57	P73/AIN0_ /T0INA/P/$\overline{D}$
56	P72/AIN2 /CLK0OUT /TX0CKIN /BSL_EN1
55	P71/AIN1/T0INB /RD4RDY/SDI
54	P70/AIN0 /RX0CKIN//WDIN /BSH_EN1 /WR4RDY
53	P97/INT6/SDO /$\overline{RD4STB}$
52	P96/INT2/SCK /WR4STB
51	P95/S0IN /$\overline{BUSACK}$
50	P94/S0OUT
49	P93/T0OUTA /RX1CKIN
48	P92/TX1CKIN /CLK1OUT
47	P91/T0OUTB /S1IN
46	P90/S1OUT
45	P27/BS7
44	P26/BS6
43	P25/BS5
42	P24/BS4
41	P23/BS3

Pin	Name
80	$\overline{RESET}$
79	OSCIN
78	V_{SS}
77	OSCOUT
76	P54/T1OUTA
75	P55/T1OUTB
74	P56/T3OUTA
73	P57/T3OUTB
72	V_{DD}
71	P80/T3INA /WR5RDY
70	P81/T1INB /RD5RDY
69	P82/T1OUTA_ /INT1/$\overline{WR5STB}$
68	P83/P/$\overline{D}$/T3INB
67	P84/T1INA /$\overline{WAIT}$/WDOUT
66	P85/P/$\overline{D}$/T3INB
65	P86/INT7 /T3OUTA

Figure 1-2. 80 Pin PQFP Package

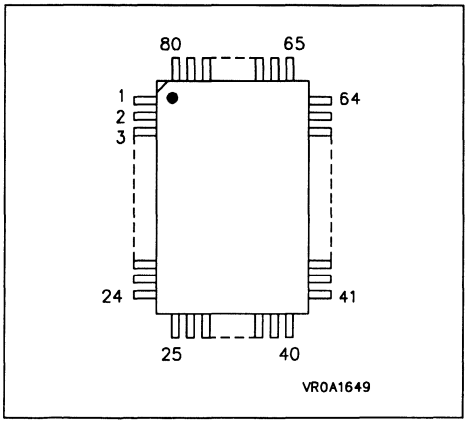


Table 3. ST90R51 Pin Description

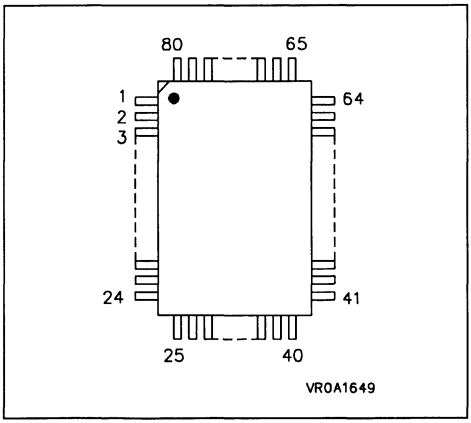
Pin	Name
1	P10/A8
2	P11/A9
3	P12/A10
4	P13/A11
5	P14/A12
6	P15/A13
7	P16/A14
8	P17/A15
9	P60/A0
10	P61/A1
11	P62/A2
12	P63/A3
13	P64/A4
14	P65/A5
15	P66/A6
16	P67/A7
17	P00/A0/D0
18	P01/A1/D1
19	P02/A2/D2
20	P03/A3/D3
21	P04/A4/D4
22	P05/A5/D5
23	P06/A6/D6
24	V_{SS}

Pin	Name
25	P07/A7/D7
26	V_{DD}
27	$\overline{AS}$
28	$\overline{DS}$
29	R/$\overline{W}$
30	P41
31	P42
32	P43
33	P44/INT0 /WDOUT
34	P45/INT4 /$\overline{BUSACK}$
35	P46/INT5 /T0OUTB
36	P47/T0OUTA /$\overline{BUSREQ}$
37	P20/BS0
38	P21/BS1
39	P22/BS2
40	P23/BS3

Pin	Name
64	P84/T1INA /$\overline{WAIT}$/WDOUT
63	P85/P/$\overline{D}$/T3INB
62	P86/INT7 /T3OUTA
61	P87/NMI /T3OUTB
60	P77
59	P76
58	P75
57	P74
56	P73/T0INA/P/$\overline{D}$
55	P72/CLK0OUT /TX0CKIN /BSL_EN1
54	P71/T0INB /RD4RDY/SDI
53	P70/RX0CKIN /WDIN /BSH_EN1 /WR4RDY
52	P97/INT6/SDO /RD4STB
51	P96/INT2/SCK /WR4STB
50	P95/S0IN /$\overline{BUSACK}$
49	P94/S0OUT
48	P93/T0OUTA /RX1CKIN
47	P92/TX1CKIN /CLK1OUT
46	P91/T0OUTB /S1IN
45	P90/S1OUT
44	P27/BS7
43	P26/BS6
42	P25/BS5
41	P24/BS4

Pin	Name
80	$\overline{RESET}$
79	OSCIN
78	V_{SS}
77	OSCOUT
76	P51
75	P52
74	P53/P/$\overline{D}$
73	P54/T1OUTA
72	P55/T1OUTB
71	P56/T3OUTA
70	P57/T3OUTB
69	V_{DD}
68	P80/T3INA /WR5RDY
67	P81/T1INB /RD5RDY
66	P82/T1OUTA /INT1/WR5STB
65	P83/T1OUTB /INT3/$\overline{RD5STB}$

Figure 1-3. 80 Pin PQFP Package



1.1 GENERAL DESCRIPTION

The ST90R50 and ST90R51, further mentioned as ST90R50 unless differently specified, are Romless members of the ST9 family of microcontrollers, completely developed and produced by SGS-THOMSON Microelectronics using a proprietary n-well HCMOS process.

The Romless part may be used for the prototyping and pre-production phases of development, and offers the maximum in program flexibility

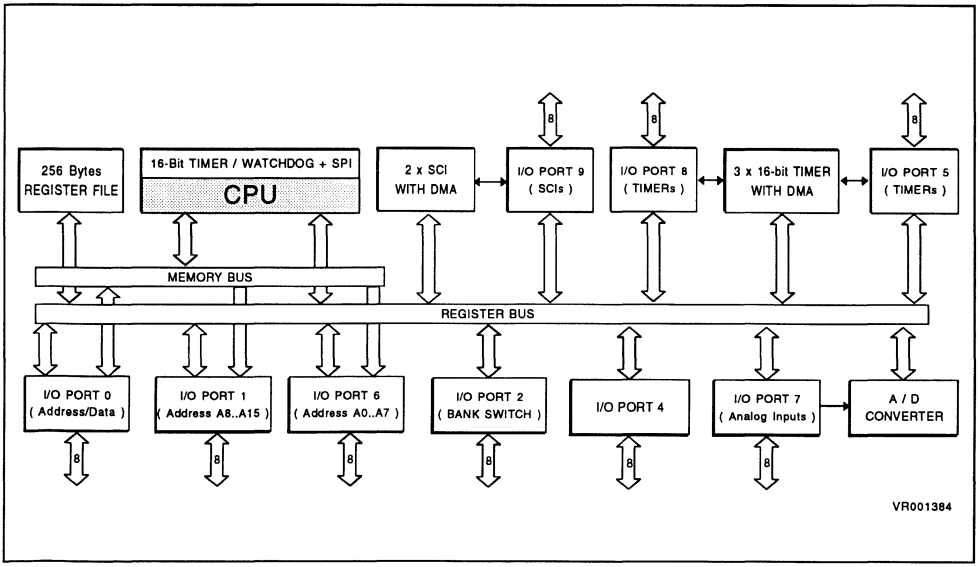
The nucleus of the ST90R50 is the advanced Core which includes the Central Processing Unit (CPU), the Register File, a 16 bit Timer/Watchdog with 8 bit Prescaler, a Serial Peripheral Interface supporting S-bus, I^2C-bus and IM-bus Interface, plus two 8 bit I/O ports. The Core has independent memory and register buses allowing a high degree of pipelining to add to the efficiency of the code execution speed of the extensive instruction set. The powerful I/O capabilities demanded by microcontroller applications are fulfilled by the ST90R50 with 56 (52 for the ST90R51) I/O lines dedicated to digital Input/Output. These lines are grouped into up to nine 8 bit I/O Ports and can be configured on a bit basis under software control to provide timing, status signals, an address/data bus for interfacing to the external memory, timer inputs and outputs, analog inputs, external interrupts and serial or parallel I/O with or without handshake.

Three basic memory spaces are available to support this wide range of configurations: Program Memory, Data Memory and the internal Register File, which includes the control and status registers of the on-chip peripherals.

Three 16 bit MultiFunction Timers, each with an 8 bit Prescaler and 13 operating modes allow simple use for complex waveform generation and measurement, PWM functions and many other system timing functions by the usage of the two associated DMA channels for each timer. In addition, for the ST90R50 only, there is an 8 channel Analog to Digital Converter with integral sample and hold, fast 11µs conversion time and 8 bit resolution. An Analog Watchdog feature is included for two input channels. For convenience the A/D converter will be noted in the following text as (R50 only).

Completing the device are 2 full duplex Serial Communications Interfaces with an integral 110 to 375,000 baud rate generator, asynchronous and 1.5Mbyte/s synchronous capability (fully programmable format) and associated address/wake-up option, plus two DMA channels.

Figure 1-4. ST90R50 Block Diagram



Note. A/D Converter for ST90R50 only

1.2 PIN DESCRIPTION

$\overline{AS}$. *Address Strobe (output, active low, 3-state).* Address Strobe is pulsed low once at the beginning of each memory cycle. The rising edge of $\overline{AS}$ indicates that address, Read/Write (R/$\overline{W}$), and Data Memory signals are valid for program or data memory transfers. Under program control, $\overline{AS}$ can be placed in a high-impedance state along with Port 0 and Port 1, Data Strobe ($\overline{DS}$) and R/$\overline{W}$.

$\overline{DS}$. *Data Strobe (output, active low, 3-state).* Data Strobe provides the timing for data movement to or from Port 0 for each memory transfer. During a write cycle, data out is valid at the leading edge of $\overline{DS}$. During a read cycle, Data In must be valid prior to the trailing edge of $\overline{DS}$. When the ST90R50 accesses on-chip memory, $\overline{DS}$ is held high during the whole memory cycle. It can be placed in a high impedance state along with Port 0, Port 1, $\overline{AS}$ and R/$\overline{W}$.

R/$\overline{W}$. *Read/Write (output, 3-state).* Read/Write determines the direction of data transfer for external memory transactions. R/$\overline{W}$ is low when writing to external program or data memory, and high for all other transactions. It can be placed in a high impedance state along with Port 0, Port 1, Port 6, $\overline{AS}$ and $\overline{DS}$.

$\overline{RESET}$. *Reset (input, active low).* The ST9 is initialised by the Reset signal. With the deactivation of $\overline{RESET}$, program execution begins from the Program memory location pointed to by the vector contained in program memory locations 00h and 01h.

OSCIN, OSCOUT. *Oscillator (input and output).* These pins connect a parallel-resonant crystal (24MHz maximum), or an external source to the on-chip clock oscillator and buffer. OSCIN is the input of the oscillator inverter and internal clock generator; OSCOUT is the output of the oscillator inverter.

AV_{DD}. Analog V_{DD} of the Analog to Digital Converter (R50 only).

AV_{SS}. Analog V_{SS} of the Analog to Digital Converter.(R50 only)

V_{DD}. Main Power Supply Voltage (5V ± 10%)

V_{SS}. Digital Circuit Ground.

P0.0-P0.7, P1.0-P1.7, P6.0-P6.7 *(Input/Output, TTL or CMOS compatible).* 8 lines grouped into I/O ports of 8 bits providing the external memory interface to address the external program memory.

P2.0-P2.7, P4.0-P4.7, P5.0-P5.7, P7.0-P7.7 (R50)
P2.0-P2.7, P4.1-P4.7, P5.1-P5.7 (R51)
P8.0-P8.7, P9.0-P9.7
I/O Port Lines (Input/Output, TTL or CMOS compatible). 8 lines grouped into I/O ports of 8 bits, bit programmable under program control as general purpose I/O or as alternate functions.

1.2.1 I/O Port Alternate Functions

Each pin of the I/O ports of the ST90R50 may assume software programmable Alternative Functions as shown in the Pin Configuration Drawings.

PIN DESCRIPTION (Continued)

Table 4. ST90R50, R51 I/O Port Alternate Function Summary

I/O PORT	Name	Function IN/OUT	Alternate Function	Pin Number		
				R50		R51
Port.bit				PLCC	PQFP	
P0.0	A0/D0	I/O	Address/Data bit 0 mux	27	17	17
P0.1	A1/D1	I/O	Address/Data bit 1 mux	28	18	18
P0.2	A2/D2	I/O	Address/Data bit 2 mux	29	19	19
P0.3	A3/D3	I/O	Address/Data bit 3 mux	30	20	20
P0.4	A4/D4	I/O	Address/Data bit 4 mux	31	21	21
P0.5	A5/D5	I/O	Address/Data bit 5 mux	32	22	22
P0.6	A6/D6	I/O	Address/Data bit 6 mux	33	23	23
P0.7	A7/D7	I/O	Address/Data bit 7 mux	35	25	25
P1.0	A8	O	Address bit 8	11	1	1
P1.1	A9	O	Address bit 9	12	2	2
P1.2	A10	O	Address bit 10	13	3	3
P1.3	A11	O	Address bit 11	14	4	4
P1.4	A12	O	Address bit 12	15	5	5
P1.5	A13	O	Address bit 13	16	6	6
P1.6	A14	O	Address bit 14	17	7	7
P1.7	A15	O	Address bit 15	18	8	8
P2.0	BS0	O	Bank Switch Address 0 (A16)	48	38	37
P2.1	BS1	O	Bank Switch Address 1 (A17)	49	39	38
P2.2	BS2	O	Bank Switch Address 2 (A18)	50	40	39
P2.3	BS3	O	Bank Switch Address 3 (A19)	51	41	40
P2.4	BS4	O	Bank Switch Address 4 (A20)	52	42	41
P2.5	BS5	O	Bank Switch Address 5 (A21)	53	43	42
P2.6	BS6	O	Bank Switch Address 6 (A22)	54	44	43
P2.7	BS7	O	Bank Switch Address 7 (A23)	55	45	44
P4.0		I/O	I/O Handshake Port 4	40	30	-
P4.1		I/O	I/O Handshake Port 4	41	31	30
P4.2		I/O	I/O Handshake Port 4	42	32	31
P4.3		I/O	I/O Handshake Port 4	43	33	32
P4.4	INT0	I	External interrupt 0	44	34	33
P4.4	WDOUT	O	T/WD output	44	34	33
P4.4		I/O	I/O Handshake Port 4	44	34	33
P4.5	INT4	I	External interrupt 4	45	35	34
P4.5	$\overline{\text{BUSACK}}$	O	External Bus Acknowledge	45	35	34

PIN DESCRIPTION (Continued)

Table 5. ST90R50, R51I/O Port Alternate Function Summary(Continued)

I/O PORT	Name	Function IN/OUT	Alternate Function	Pin Number		
				R50		R51
Port.bit				PLCC	PQFP	
P4.5		I/O	I/O Handshake Port 4	45	35	34
P4.6	INT5	I	External interrupt 5	46	36	35
P4.6	T0OUTB	O	MF Timer 0 output B	46	36	35
P4.6		I/O	I/O Handshake Port 4	46	36	35
P4.7	T0OUTA	O	MF Timer 0 output A	47	37	36
P4.7	$\overline{\text{BUSREQ}}$	I	External Bus Request	47	37	36
P4.7		I/O	I/O Handshake Port 4	47	37	36
P5.0		I/O	I/O Handshake Port 5	6	-	-
P5.1		I/O	I/O Handshake Port 5	5	-	76
P5.2		I/O	I/O Handshake Port 5	4	-	75
P5.3		I/O	I/O Handshake Port 5	3	-	74
P5.3	P/$\overline{\text{D}}$	O	Program/Data space select	3	-	74
P5.4	T1OUTA	O	MF Timer 1 output A	2	76	73
P5.4		I/O	I/O Handshake Port 5	2	76	73
P5.5	T1OUTB	O	MF Timer 1 output B	1	75	72
P5.5		I/O	I/O Handshake Port 5	1	75	72
P5.6	T3OUTA	O	MF Timer 3 output A	84	74	71
P5.6		I/O	I/O Handshake Port 5	84	74	71
P5.7	T3OUTB	O	MF Timer 3 output B	83	73	70
P5.7		I/O	I/O Handshake Port 5	83	73	70
P6.0	A0	O	Address bit 0 (non mux)	19	9	9
P6.1	A1	O	Address bit 1 (non mux)	20	10	10
P6.2	A2	O	Address bit 2 (non mux)	21	11	11
P6.3	A3	O	Address bit 3 (non mux)	22	12	12
P6.4	A4	O	Address bit 4 (non mux)	23	13	13
P6.5	A5	O	Address bit 5 (non mux)	24	14	14
P6.6	A6	O	Address bit 6 (non mux)	25	15	15
P6.7	A7	O	Address bit 7 (non mux)	26	16	16
P7.0	AIN0	I	A/D Analog input 0 (R50 only)	64	54	
P7.0	ADTRG	I	A/D conversion trigger	64	54	53
P7.0	WRRDY4	O	Handshake Write Ready P4	64	54	53
P7.0	RX0CKIN	I	SCI0 Receive Clock input	64	54	53
P7.0	WDIN	I	T/WD input	64	54	53

PIN DESCRIPTION (Continued)

Table 6. ST90R50, R51 I/O Port Alternate Function Summary(Continued)

I/O PORT	Name	Function IN/OUT	Alternate Function	Pin Number		
				R50		R51
Port.bit				PLCC	PQFP	
P7.0	BSH_EN1	I	Bank Switch High Nibble Enable	64	54	53
P7.1	AIN1	I	A/D Analog input 1 (R50 only)	65	55	
P7.1	RDRDY4	O	Handshake Read Ready P4	65	55	54
P7.1	SDI	I	SPI Serial Data In	65	55	54
P7.1	T0INB	I	MF Timer 0 input B	65	55	54
P7.2	AIN2	I	A/D Analog input 2 (R50 only)	66	56	
P7.2	CLK0OUT	O	SCI0 Byte Sync Clock output	66	56	55
P7.2	TX0CKIN	I	SCI0 Transmit Clock input	66	56	55
P7.2	BSL_EN1	I	Bank Switch Low Nibble Enable	66	56	55
P7.3	AIN3	I	A/D Analog input 3 (R50 only)	67	57	
P7.3	P/$\overline{D}$	O	Program/data space select	67	57	56
P7.3	T0INA	I	MF Timer 0 input A	67	57	56
P7.4	AIN4	I	A/D Analog input 4 (R50 only)	68	58	57
P7.5	AIN5	I	A/D Analog input 5 (R50 only)	69	59	58
P7.6	AIN6	I	A/D Analog input 6 (R50 only)	70	60	59
P7.7	AIN7	I	A/D Analog input 7 (R50 only)	71	61	60
P8.0	WRRDY5	O	Handshake Write Ready P5	81	71	68
P8.0	T3INA	I	MF Timer 3 input A	81	71	68
P8.1	RDRDY5	O	Handshake Read Ready P5	80	70	67
P8.1	T1INB	I	MF Timer 1 input B	80	70	67
P8.2	INT1	I	External interrupt 1	79	69	66
P8.2	T1OUTA	O	MF Timer 1 output A	79	69	66
P8.2	$\overline{WRSTB5}$	O	Handshake Write Strobe P5	79	69	66
P8.3	INT3	I	External interrupt 3	78	68	65
P8.3	T1OUTB	O	MF Timer 1 output B	78	68	65
P8.3	$\overline{RDSTB5}$	I	Handshake Read Strobe P5	78	68	65
P8.4	T1INA	I	MF Timer 1 input A	77	67	64
P8.4	$\overline{WAIT}$	I	External Wait input	77	67	64
P8.4	WDOUT	O	T/WD output	77	67	64
P8.5	P/$\overline{D}$	O	Program/Data space select	76	66	63
P8.5	T3INB	I	MF Timer 3 input B	76	66	63
P8.6	INT7	I	External interrupt 7	75	65	62

PIN DESCRIPTION (Continued)

Table 2. ST90R50, R51 I/O Port Alternate Function Summary(Continued)

I/O PORT	Name	Function IN/OUT	Alternate Function	Pin Number		
				R50		R51
Port.bit				PLCC	PQFP	
P8.6	T3OUTA	O	MF Timer 3 output A	75	65	62
P8.7	NMI	I	Non-Maskable Interrupt	74	64	61
P8.7	T3OUTB	O	MF Timer 3 output B	74	64	61
P9.0	S1OUT	O	SCI1 Serial Output	56	46	45
P9.1	T0OUTB	O	MF Timer 0 output B	57	47	46
P9.1	S1IN	I	SCI1 Serial Input	57	47	46
P9.2	CLK1OUT	O	SCI1 Byte Sync Clock output	58	48	47
P9.2	TX1CKIN	I	SCI1 Transmit Clock input	58	48	47
P9.3	RX1CKIN	I	SCI1 Receive Clock input	59	49	48
P9.3	T0OUTA	O	MF Timer 0 output A	59	49	48
P9.4	S0OUT	O	SCI0 Serial Output	60	50	49
P9.5	S0IN	I	SCI0 Serial Input	61	51	50
P9.5	$\overline{\text{BUSACK}}$	O	External Bus Acknowledge	61	51	50
P9.6	INT2	I	External interrupt 2	62	52	51
P9.6	SCK	O	SPI Serial Clock	62	52	51
P9.6	$\overline{\text{WRSTB4}}$	O	Handshake Write Strobe P4	62	52	51
P9.7	INT6	I	External interrupt 6	63	53	52
P9.7	SDO	O	SPI Serial Data Out	63	53	52
P9.7	$\overline{\text{RDSTB4}}$	I	Handshake Read Strobe P4	63	53	52

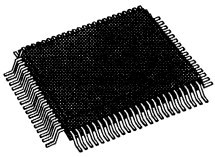
SGS-THOMSON MICROELECTRONICS

ST90R91

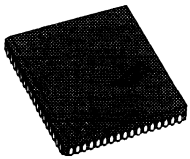
ROMLESS HCMOS MCU WITH BANKSWITCH AND A/D CONVERTER

PRELIMINARY DATA

- Register oriented 8/16 bit CORE with RUN, WFI and HALT modes
- Minimum instruction cycle time : 500ns (12MHz internal)
- ROMless to allow maximum external memory flexibility
- 1536 bytes of internal RAM
- 224 general purpose registers available as RAM, accumulators or index pointers (register file)
- Bankswitch logic allowing a maximum addressing capability of up to 2Mbytes for Program and Dataspace
- 80-pin PQFP package for ST90R91Q
- 68-pin PLCC package for ST90R91C
- DMA controller, Interrupt handler and a Serial Peripheral Interface as standard features
- Up to 32 fully programmable I/O ports
- Up to 7 external plus 1 non-maskable interrupts
- 16-bit Timer with 8 bit Prescaler, able to be used as a Watchdog Timer
- 16-bit Slice Timer with 8 bit Prescaler
- 16-bit Multifunction Timer, with 8 bit prescaler and 13 operating modes
- 4 channel 8 bit Analog to Digital Converter, with Analog Watchdogs and external references
- Rich Instruction Set with 14 Addressing modes
- Division-by-Zero trap generation
- Versatile Development Tools, including assembler, linker, C-compiler, archiver, graphic oriented debugger and hardware emulators
- Real Time Operating System



PQFP80



PLCC68

(Ordering Information at the end of the Datasheet)

Figure 1. 80 Pin PQFP Package

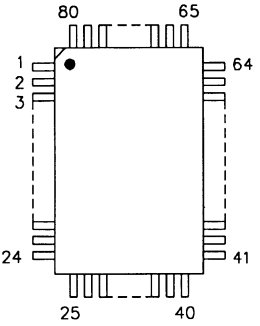


Table 1. ST90R91Q Pin Description

Pin	Name
1	NC
2	P62/A2
3	P61/A1
4	P60/A0
5	P23/BS3
6	P22/BS2
7	P21/BS1
8	P20/BS0
9	P07/AD7
10	P06/AD6
11	P05/AD5
12	P00/AD0
13	P04/AD4
14	P01/AD1
15	P03/AD3
16	P02/AD2
17	V_{CC}
18	$\overline{AS}$
19	$\overline{DS}$
20	R/$\overline{W}$
21	NC
22	NC
23	P50/W/$\overline{R}$
24	P51

Pin	Name
25	P52
26	P53/P/$\overline{D}$
27	P54
28	P55
29	P56
30	P57
31	P40
32	P41/WDIN
33	P42/WDOUT
34	P43/T0INPA
35	P44/T0INPB
36	P45/T0OUTA /ADTRG
37	P46/T0OUTB
38	P47/AIN4
39	P80/AIN5
40	NC

Pin	Name
64	P75/SDO
63	P74/INT6 /BSH_EN1/SLOUT
62	P73
61	NC
60	P72/INT3 /SLIN
59	P71/INT7
58	P70/INT1/$\overline{WAIT}$
57	P27/BS7
56	P26/BS6
55	P25/BS5
54	P24/BS4
53	P87/NMI
52	P86/INT5
51	P85/INT4
50	P84
49	P83/T0OUTA
48	V_{SS2}
47	$\overline{RESET}$
46	OSCIN
45	V_{SS}
44	OSCOUT
43	P82/AIN7
42	P81/AIN6
41	NC

Pin	Name
80	NC
79	P63/A3
78	P64/A4
77	P65/A5
76	P66/A6
75	P67/A7
74	P12/A10
73	P13/A11
72	P11/A9
71	P10/A8
70	P15/A13
69	P14/A12
68	P16/A14
67	P17/A15
66	P77/SDO/SDI
65	P76/SCK/INT2 /BSL_EN1

Note. NC = Not Connected

Figure 2. 68 Pin PLCC Package

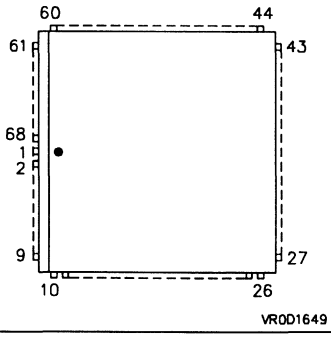


Table 2. ST90R91C Pin Description

Pin	Name
61	P74/INT6 /BSH_EN1/SLOUT
62	P75/SDO
63	P76/SCK/INT2 /BSL_EN1
64	P77/SDO/SDI
65	P17/A15
66	P16/A14
67	P14/A12
68	P15/A13
● 1	P10/A8
2	P11/A9
3	P13/A11
4	P12/A10
5	P67/A7
6	P66/A6
7	P65/A5
8	P64/A4
9	P63/A3

Pin	Name
10	P62/A2
11	P61/A1
12	P60/A0
13	P21/BS1
14	P20/BS0
15	P07/AD7
16	P06/AD6
17	P05/AD5
18	P00/AD0
19	P04/AD4
20	P01/AD1
21	P03/AD3
22	P02/AD2
23	V_{CC}
24	$\overline{AS}$
25	$\overline{DS}$
26	R/$\overline{W}$

Pin	Name
43	P80/AIN5
42	P47/AIN4
41	P46/T0OUTB
40	P45/T0OUTA /ADTRG
39	P44/T0INPB
38	P43/T0INPA
37	P42/WDOUT
36	P41/WDIN
35	P40
34	P57
33	P56
32	P55
31	P54
30	P53/P/$\overline{D}$
29	P52
28	P51
27	P50/W/$\overline{R}$

Pin	Name
60	P72/INT3/SLIN
59	P71/INT7
58	P70/INT1/WAIT
57	P25/BS5
56	P24/BS4
55	P87/NMI
54	P86/INT5
53	P85/INT4
52	P84
51	P83/T0OUTA
50	V_{SS2}
49	$\overline{RESET}$
48	OSCIN
47	V_{SS}
46	OSCOUT
45	P82/AIN7
44	P81/AIN6

1.1 GENERAL DESCRIPTION

The ST90R91 is a Romless member of the ST9 family of microcontrollers, completely developed and produced by SGS-THOMSON Microelectronics using a proprietary n-well HCMOS process.

The Romless part may be used for the prototyping and pre-production phases of development, and offers the maximum in program flexibility

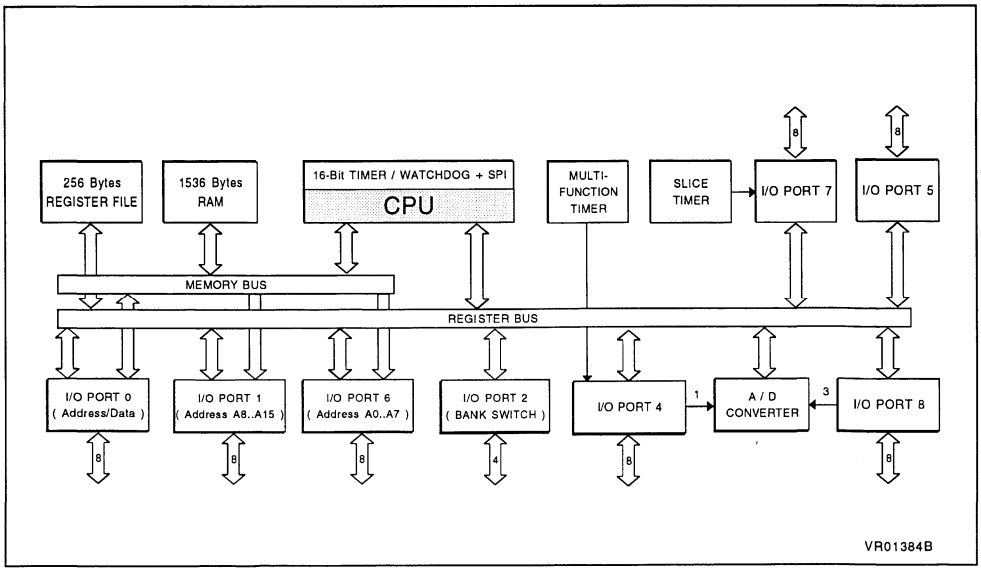
The nucleus of the ST90R91 is the advanced Core which includes the Central Processing Unit (CPU), the Register File, a 16 bit Timer/Watchdog with 8 bit Prescaler, a Serial Peripheral Interface supporting S-bus, I^2C-bus and IM-bus Interface, plus two 8 bit I/O ports. The Core has independent memory and register buses allowing a high degree of pipelining to add to the efficiency of the code execution speed of the extensive instruction set. The powerful I/O capabilities demanded by microcontroller applications are fulfilled by the ST90R91 with up to 32 I/O lines dedicated to digital Input/Output. These lines are grouped into nine 8 bit I/O Ports and can be configured on a bit basis under software control to provide timing, status signals, an address/data bus for interfacing to the external memory, timer inputs and outputs, analog inputs, external interrupts and serial or parallel I/O.

Three basic memory spaces are available to support this wide range of configurations: Program Memory , Data Memory and the internal Register File, which includes the control and status registers of the on-chip peripherals.

A 16 bit MultiFunction Timer, with an 8 bit Prescaler and 13 operating modes allows simple use for complex waveform generation and measurement, PWM functions and many other system timing functions by the usage of the two associated DMA channels for each timer. In addition there is an 4 channel Analog to Digital Converter with integral sample and hold, fast 11µs conversion time and 8 bit resolution. An Analog Watchdog feature is included for two input channels.

Completing the device is the Slice Timer, capable of generating PWM signals and simple timing functions.

Figure 1-3. ST90R91 Block Diagram



1.2 PIN DESCRIPTION

$\overline{AS}$. *Address Strobe (output, active low, 3-state).* Address Strobe is pulsed low once at the beginning of each memory cycle. The rising edge of $\overline{AS}$ indicates that address, Read/Write (R/$\overline{W}$), and Data Memory signals are valid for program or data memory transfers. Under program control, $\overline{AS}$ can be placed in a high-impedance state along with Port 0 and Port 1, Data Strobe ($\overline{DS}$) and R/$\overline{W}$.

$\overline{DS}$. *Data Strobe (output, active low, 3-state).* Data Strobe provides the timing for data movement to or from Port 0 for each memory transfer. During a write cycle, data out is valid at the leading edge of $\overline{DS}$. During a read cycle, Data In must be valid prior to the trailing edge of $\overline{DS}$. When the ST90R91 accesses on-chip memory, $\overline{DS}$ is held high during the whole memory cycle. It can be placed in a high impedance state along with Port 0, Port 1, $\overline{AS}$ and R/$\overline{W}$.

R/$\overline{W}$. *Read/Write (output, 3-state).* Read/Write determines the direction of data transfer for external memory transactions. R/$\overline{W}$ is low when writing to external program or data memory, and high for all other transactions. It can be placed in a high impedance state along with Port 0, Port 1, Port 6, $\overline{AS}$ and $\overline{DS}$.

$\overline{RESET}$. *Reset (input, active low).* The ST9 is initialised by the Reset signal. With the deactivation of $\overline{RESET}$, program execution begins from the Program memory location pointed to by the vector contained in program memory locations 00h and 01h.

OSCIN, OSCOUT. *Oscillator (input and output).* These pins connect a parallel-resonant crystal (24MHz maximum), or an external source to the on-chip clock oscillator and buffer. OSCIN is the input of the oscillator inverter and internal clock generator; OSCOUT is the output of the oscillator inverter.

V_{DD}. Main Power Supply Voltage (5V ± 10%)

V_{SS}., V_{SS2} Digital Circuit Ground.

P0.0-P0.7, P1.0-P1.7, P6.0-P6.7 *(Input/Output, TTL or CMOS compatible).* 8 lines grouped into I/O ports of 8 bits providing the external memory interface to address the external program memory.

P2.0-P2.7, P4.0-P4.7, P5.0-P5.7, P7.0-P7.7, P8.0-P8.7 *I/O Port Lines (Input/Output, TTL or CMOS compatible).* 8 lines grouped into I/O ports of 8 bits, bit programmable under program control as general purpose I/O or as alternate functions.

1.2.1 I/O Port Alternate Functions

Each pin of the I/O ports of the ST90R91 may assume software programmable Alternative Functions as shown in the Pin Configuration Drawings. Table 3 shows the Functions allocated to each I/O Port pins.

PIN DESCRIPTION (Continued)

Table 3. ST90R91 I/O Port Alternate Function Summary

I/O PORT	Name	Function IN/OUT	Alternate Function	Pin Assignment	
Port.bit				PLCC	PQFP
P0.0	A0/D0	I/O	Address/Data bit 0 mux	18	12
P0.1	A1/D1	I/O	Address/Data bit 1 mux	20	14
P0.2	A2/D2	I/O	Address/Data bit 2 mux	22	16
P0.3	A3/D3	I/O	Address/Data bit 3 mux	21	15
P0.4	A4/D4	I/O	Address/Data bit 4 mux	19	13
P0.5	A5/D5	I/O	Address/Data bit 5 mux	17	11
P0.6	A6/D6	I/O	Address/Data bit 6 mux	16	10
P0.7	A7/D7	I/O	Address/Data bit 7 mux	15	9
P1.0	A8	O	Address bit 8	1	71
P1.1	A9	O	Address bit 9	2	72
P1.2	A10	O	Address bit 10	4	74
P1.3	A11	O	Address bit 11	3	73
P1.4	A12	O	Address bit 12	67	69
P1.5	A13	O	Address bit 13	68	70
P1.6	A14	O	Address bit 14	66	68
P1.7	A15	O	Address bit 15	65	67
P2.0	BS0	O	Bank Switch Address 0	14	8
P2.1	BS1	O	Bank Switch Address 1	13	7
P2.2	BS2	O	Bank Switch Address 2	-	6
P2.3	BS3	O	Bank Switch Address 3	-	5
P2.4	BS4	O	Bank Switch Address 4	56	54
P2.5	BS5	O	Bank Switch Address 5	57	55
P2.6	BS6	O	Bank Switch Address 6	-	56
P2.7	BS7	O	Bank Switch Address 7	-	57
P4.0		I/O		35	31
P4.1	WDIN	I	T/WD input	36	32
P4.2	WDOUT	O	T/WD output	37	33
P4.3	T0INA	I	MF Timer 0 Input A	38	34
P4.4	T0INB	I	MF Timer 0 Input B	39	35
P4.5	T0OUTA	O	MF Timer 0 output A	40	36
P4.5	ADTRG	I	A/D Conversion Trigger	40	36
P4.6	T0OUTB	O	MF Timer 0 output B	41	37
P4.7	AIN4	I	A/D Analog Input 4	42	38

PIN DESCRIPTION (Continued)

Table 3. ST90R91 I/O Port Alternate Function Summary(Continued)

I/O PORT	Name	Function IN/OUT	Alternate Function	Pin Assignment	
Port.bit				PLCC	PQFP
P5.0	W/$\overline{R}$	O	Write/$\overline{Read}$ Strobe	27	23
P5.1		I/O		28	24
P5.2		I/O		29	25
P5.3	P/$\overline{D}$	O	Program/Data Space Select	30	26
P5.4		I/O		31	27
P5.5		I/O		32	28
P5.6		I/O		33	29
P5.7		I/O		34	30
P6.0	A0	O	Address bit 0 (non mux)	12	4
P6.1	A1	O	Address bit 1 (non mux)	11	3
P6.2	A2	O	Address bit 2 (non mux)	10	2
P6.3	A3	O	Address bit 3 (non mux)	9	79
P6.4	A4	O	Address bit 4 (non mux)	8	78
P6.5	A5	O	Address bit 5 (non mux)	7	77
P6.6	A6	O	Address bit 6 (non mux)	6	76
P6.7	A7	O	Address bit 7 (non mux)	5	75
P7.0	INT1	I	External Interrupt 1	58	58
P7.0	$\overline{WAIT}$	I	External Wait Input	58	58
P7.1	INT7	I	External Interrupt 7	59	59
P7.2	INT3	I	External Interrupt 3	60	60
P7.2	SLIN	I	Slice Timer Input	60	60
P7.3		I/O		-	62
P7.4	INT6	I	External Interrupt 6	61	63
P7.4	BSH_EN1	I	Bankswitch High Nibble Enable	61	63
P7.4	SLOUT	O	Slice Timer Output	61	63
P7.5	SDO	O	SPI Serial Data Output	62	64
P7.6	SCK	O	SPI Serial Clock	63	65
P7.6	INT2	I	External Interrupt 2	63	65
P7.6	BSL_EN1	I	Bankswitch Low Nibble Enable	63	65
P7.7	SDO	O	SPI Serial Data Output	64	66
P7.7	SDI	I	SPI Serial Data Input	64	66
P8.0	AIN5	I	A/D Analog Input 5	43	39

PIN DESCRIPTION (Continued)

Table 3. ST90R91 I/O Port Alternate Function Summary(Continued)

I/O PORT	Name	Function IN/OUT	Alternate Function	Pin Assignment	
Port.bit				PLCC	PQFP
P8.1	AIN6	I	A/D Analog Input 6	44	42
P8.2	AIN7	I	A/D Analog Input 7	45	43
P8.3	T0OUTA	O	MF Timer 0 output A	51	49
P8.4		I/O		52	50
P8.5	INT4	I	External interrupt 4	53	51
P8.6	INT5	I	External interrupt 5	54	52
P8.7	NMI	I	Non-Maskable Interrupt	55	53

ST9 INSTRUCTION SET

1 SOFTWARE DESCRIPTION

1 SOFTWARE DESCRIPTION

1.1 ADDRESSING MODES

The ST9 offers a wide variety of established and new addressing modes and combinations to facilitate full and rapid access to the address spaces while reducing program length. The available addressing modes are shown in Table 1-1:

Single operand arithmetic, logic and shift byte instructions have direct register and indirect register addressing modes. For a full list of the possible combinations for each instruction type, please refer to the ST9 Programming Manual.

Table 1-1. Addressing Modes

Operand is In	Addressing Mode	Destination Location	Notation
Instruction	Immediate	Byte Word	#N #NN
Register File	Direct	Byte Word	r rr
	Indirect	Byte/Word	(r)
	Indexed	Byte/Word	N(r)
	Indirect Post-Increment	Byte	(r)+
Program or Data Memory	Direct	Byte/Word	NN
	Indirect	Byte/Word	(rr)
	Indirect Post-Increment	Byte/Word	(rr)+
	Indirect Pre-Decrement	Byte/Word	-(rr)
	Short Indexed	Byte/Word	N(rr)
	Long Indexed	Byte/Word	NN(rr)
	Register Indexed	Byte/Word	rr(rr)
Any bit of any working register	Direct	Bit	r.b
Any bit in program or data memory	Indirect	Bit	(rr).b

ADDRESSING MODES (Continued)

Two Operands Arithmetic and Logic Instructions	
Destination	**Source**
Register Direct	Register Direct
Register Direct	Register Indirect
Register Direct	Memory Indirect
Register Direct	Memory Indexed
Register Direct	Memory Indirect with Post-Increment
Register Direct	Memory Indirect with Pre-Decrement
Register Direct	Memory Direct
Register Indirect	Register Direct
Memory Indirect	Register Direct
Memory Indexed	Register Direct
Memory Indirect with Post-Increment	Register Direct
Memory Indirect with Pre-Decrement	Register Direct
Memory Direct	Register Direct
Register Direct	Immediate
Memory Direct	Immediate
Memory Indirect	Immediate

Two Operands Arithmetic, Logic and Load Instructions	
Destination	**Source**
Memory Indirect	Memory Direct

ADDRESSING MODES (Continued)

Two Operands Load Instructions	
Destination	**Source**
Register Direct	Register Direct
Register Direct	Register Indirect
Register Direct	Register Indexed
Register Direct	Memory Indirect
Register Direct	Memory Indexed
Register Direct	Memory Indirect with Post-Increment
Register Direct	Memory Indirect with Pre-Decrement
Register Direct	Memory Direct
Register Indirect	Register Direct
Register Indexed	Register Direct
Memory Indirect	Register Direct
Memory Indexed	Register Direct
Memory Indirect with Post-Increment	Register Direct
Memory Indirect with Pre-Decrement	Register Direct
Memory Direct	Register Direct
Register Direct	Immediate
Memory Direct	Immediate
Memory Indirect	Immediate
Long Indexed Memory (1)	Immediate

Two Operands Load Instructions (2)	
Destination	**Source**
Register Indirect with Post-Increment	Memory Indirect with Post-Increment
Memory Indirect with Post-Increment	Register Indirect with Post-Increment
Memory Indirect with Post-Increment	Memory Indirect with Post-Increment

Notes:
1. Word Instructions Only
2. Load Byte Only

ADDRESSING MODES (Continued)

1.1.1 Register Addressing Modes

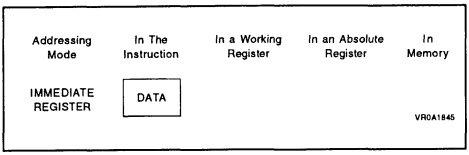
Immediate Addressing Mode

In the Immediate addressing mode, the data is found in the instruction. When using immediate data, a hash-mark (#) is used to distinguish it from an absolute address in memory.

Example: `ldw RR42,#65536`
loads the immediate value 65536 into the register pair R42 & R43. While the example shows decimal data, hexadecimal and binary values may also be used.

Example: `ldw RR42,#0FFFFh.`

Figure 1-1. Immediate Register

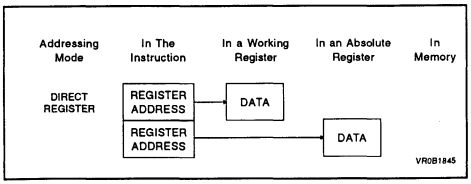


Direct Addressing Mode

In the direct addressing mode, a register can be addressed by using its absolute address in the Register File (in decimal, hexadecimal or binary form). Alternatively a register can be addressed directly as a working register;

Example: `xch R0A2h,r4`
exchanges the values in the register RA2h and working register number 4.

Figure 1-2. Direct Register



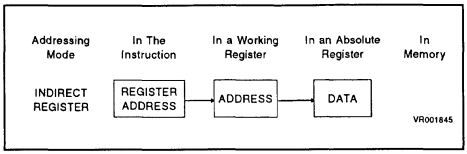
Indirect Addressing Mode

In the Indirect Register Addressing mode, the address of the data does not appear in the instruction but is located in a working register. The address of this register is given in the instruction. The indirect addressing mode is indicated by the use of parentheses.

Example:
If register 200 contains 178 and working register 11 contains 86 then the instruction `ld (r11),R200` loads the value 178 into register 86.

Note: the indirect address can only be contained in a working register.

Figure 1-3. Indirect Register



Indexed Addressing Mode

To address a register using the Indexed mode, an offset value is used to add to an index value (which acts as a base or starting value). The offset value is the Immediate value given in the instruction while the index value is given by the contents of the working register.

Example: if working register 10 contains 55 then the instruction

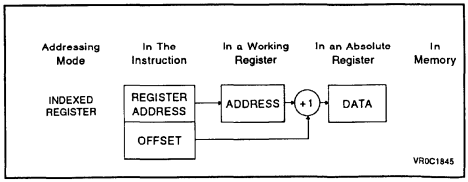
ld 40(r10),r18

loads register 95 (i.e.55+40) with the contents of working register 18.

The Register File never needs an absolute value requiring more than one byte and therefore only requires a short offset and a single register to contain the index.

Note: The index value can only be contained in a working register.

Figure 1-4. Indexed Register



ADDRESSING MODES (Continued)

Indirect Register Post-increment Addressing Mode

In this addressing mode, both destination and source addresses are given by the contents of working registers which are then post-incremented. The address of the memory location is contained in a working register pair, and the address of the register is contained into a single working register. Only working registers may be used to contain the addresses, this mode being indicated by both source and destination using parentheses followed by plus sign.

Example: if working register 8 contains the value 44, working register pair rr2 contains the value 2000, and register 44 contains the value 56, then by using the instruction

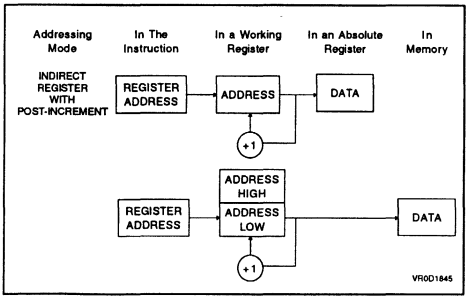
ld (rr2)+,(r8)+

the memory location 2000 will be loaded with the value 56. Immediately following this, the contents of r8 is incremented to 45 and the contents of rr2 is incremented to 2001.

This addressing mode is useful for moving blocks of data either from Register File to Memory or from Memory to Register File.

Direct Bit Addressing Mode

Figure 1-5. Register Indirect Post-Increment



In the direct bit addressing mode, any bit in any working register can be addressed

Examples: `bset r7.3`

This instruction sets the bit 3 of the working register 7.

bld r7.3, r12.6

This instruction loads the bit 6 of the working register 12 in bit 3 of working register 7

1.1.2 Memory Addressing Modes

The memory addressing modes described in this section are available to data and program memory. Thus before addressing the memory, it is necessary to indicate by use of the Set Program/Data Memory instructions, `spm` and `sdm`, in which memory the instructions are working. Since each memory space is 64K byte long, a word address is necessary to specify memory locations.

Direct Addressing Mode

The Memory Direct addressing mode requires the specific location within the memory. This only needs the absolute offset value which can be given in decimal, hex or binary form.

Thus the instruction

ld 12345,r9

loads working register 9 data into memory location 12345

In the memory direct mode, it is possible to use an immediate addressing mode for the source operand.

Examples:

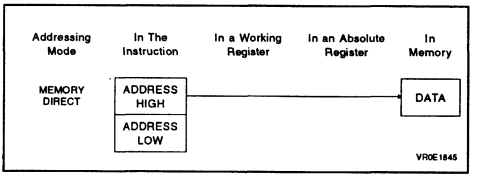
ld 12354,#34

will load the value 34 into the memory location 12354.

ldw 12354,#3457

will load the location pair 12354 and 12355 with the value 3457.

Figure 1-6. Memory Direct



ADDRESSING MODES (Continued)

Indirect Addressing Mode

When using the indirect addressing mode to access memory, the address is contained in a pair of working registers.

Example: if the working register pair r8 and r9 contains the value 2000 then the instruction

ld (rr8),#34

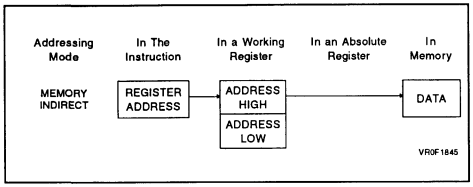
loads the value 34 into memory location 2000.

If the data to be stored is a word then the instruction `ldw` will automatically interpret the address as a pair of memory locations. So if rr8 contains 2000 then the instruction

ldw (rr8),#3467

loads the memory locations 2000 and 2001 with the value 3467.

Figure 1-7. Memory Indirect



Indirect With Post-increment Adressing Mode

The indirect with post-increment addressing mode is similar to the memory indirect addressing mode but, in addition, after accessing the data in the currently pointed address, the value in the pointing working register pair is incremented. This mode is indicated by a plus sign following a working register pair in parentheses, e.g. `(rr4)+`.

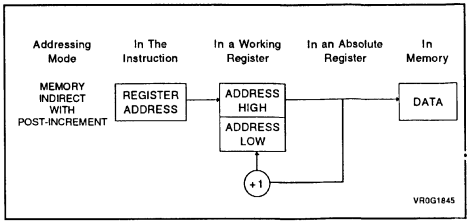
Example:
If the working register pair rr4 (working registers r4 and r5) contains the value 3000 and memory location 3000 contains the value 88, then the instruction

ld R50,(rr4)+

loads register 50 with the value 88 and then the value in rr4 to be incremented to 3001.

This mode uses only working registers to contain the address. Thus the Indirect with Post-Increment addressing mode is most useful in repeated situations when a number of adjacent items of data are required in succession. The use of this addressing mode saves both time and program memory space since it cuts the usual increment instruction.

Figure 1-8. Memory Indirect Post-Increment



Indirect With Pre-decrement Addressing Mode

This indirect memory addressing mode has an automatic pre-decrement. The address can only be contained in working registers and the mode is indicated by a minus sign in front of the working registers which are in parentheses, e.g. `-(rr6)`.

Thus if the working register pair rr6 contains the value 1111 and location 1110 contains the value 40 then the instruction

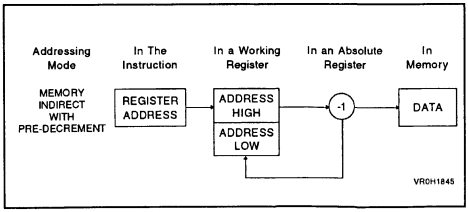
ld R56,-(rr6)

decrements the value in rr6 to 1110 and then loads the value 40 into register 56.

This addressing mode allows the ST9 to deal in the reverse order with data previously managed using the indirect post-increment mode without resetting the pointing registers (of the last post-increment).

The pre-decrement mode has the same benefits of time and program memory saving as the post-increment mode.

Figure 1-9. Memory Indirect Pre-Decrement



ADDRESSING MODES (Continued)

Indexed Addressing Modes

There are three indexed addressing modes, each using an indirect address plus offset format. The index address is given as an indirect address contained in a working register pair, while the offset can be long or short (a word or a byte). The address of the data required is given by the value of the working register pair indicated (the index), plus the value of the given offset. The specification of this offset which differentiates the three modes, is as follows:

- Indexed with an Immediate Short and Long Offset

In these indexed modes the offset is a fixed and Immediate value included in the instruction. It may be either a short or long index as required, this immediate value being added to the address given by the working register pair.

Example: if the working register pair, rr6, contains the value 8000 and memory location 8034 contains the value 254 then the instruction

ld R55,34(rr6)

loads the value 254 into register 55.

Or, as another example, if the working register pair rr2 contains the value 2000 and register 78 contains the value 34 then the instruction.

ld 322(rr2),r78

loaded the value 34 into memory location 2322.

- Indexed with a Register Offset

In this addressing mode, the index is supplied by one pair of working registers and the offset is supplied by a second pair of working registers. The format is rrx(rry), x and y being in the range 0,2,4...12,14.

Example
If working register pair rr0 contains the value 2222 and working register pair rr4 contains 3333 while register 45 contains the value 78 then the instruction

ld rr4(rr0),R45

loads the value 78 into memory location 5555.

Figure 1-10. Memory Indexed with Immediate Short Offset

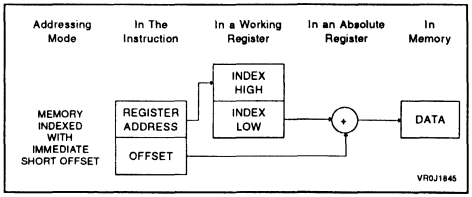


Figure 1-11. Memory Indexed with Immediate Long Offset

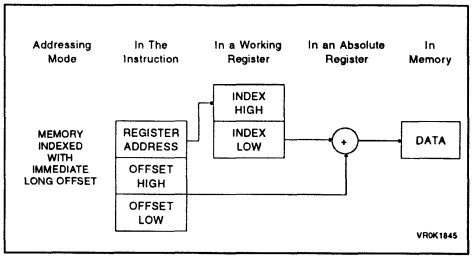
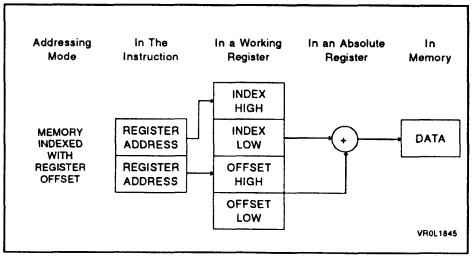


Figure 1-12. Memory Indexed with Register Offset



Indirect Memory Bit Addressing Mode

In the indirect memory bit addressing mode, any bit of Program/Data memory location can be addressed with the `btset` (Bit Test and SET) instruction.

Example

btset (rr8).3

This instruction sets bit 3 of the memory location addressed by the working registers r8, r9 contents.

1.2 INSTRUCTION SET

The ST9 instruction set consists of 87 instruction types which can be divided into eight groups:

- Load (two operands)
- Arithmetic & logic (two operands)
- Arithmetic Logic and Shift (one operand)
- Stack (one operand)
- Multiply & Divide (two operands)
- Boolean (one or two operands)
- Program Control (zero to three operands)
- Miscellaneous (zero to two operands)

The wide range of instructions eases use of the register file and address spaces, reducing operation times, while the register pointers mechanism allows an unmatched code efficiency and ultrafast context switching. A particularly notable feature is the comprehensive "Any Bit, Any Register" (ABAR) addressing capability of the Boolean instructions.

The ST9 can operate with a wide range of data lengths from single bits, 4-bit nibbles which can be in the form of Binary Coded Decimal (BCD) digits, 8-bit bytes, and 16-bit words.

The following summary shows the instructions belonging to each group and the number of operands required for each instruction. The source operand is "src", "dst" is the destination operand, and "cc" is a condition code.

INSTRUCTION SET (Continued)

Load Instructions (Two Operands)

Mnemonic	Operands	Instruction
LD	dst,src	Load
LDW	dst,src	Load Word
LDPP	dst,src	Load Program Memory -> Program Memory
LDPD	dst,src	Load Data Memory -> Program Memory
LDDP	dst,src	Load Program Memory -> Data Memory
LDDD	dst,src	Load Data Memory -> Data Memory

Arithmetic and Logic Instructions (Two Operands)

Mnemonic	Operands	Instruction
ADD	dst,src	Add
ADDW	dst,src	Add Word
ADC	dst,src	Add With carry
ADCW	dst,src	Add Word With Carry
SUB	dst,src	Substract
SUBW	dst,src	Substract Word
SBC	dst,src	Substract With Carry
SBCW	dst,src	Substract Word With Carry
AND	dst,src	Logical AND
ANDW	dst,src	Logical Word AND
OR	dst,src	Logical OR
ORW	dst,src	Logical Word OR
XOR	dst,src	Logical Exclusive OR
XORW	dst,src	Logical Word Exclusive OR
CP	dst,src	Compare
CPW	dst,src	Compare Word
TM	dst,src	Test Under Mask
TMW	dst,src	Test Word Under Mask
TCM	dst,src	Test Complement Under Mask
TCMW	dst,src	Test Word Complement Under Mask

INSTRUCTION SET (Continued)

Arithmetic Logic and Shift Instructions (One Operand)

Mnemonic	Operands	Instruction
INC INCW	dst dst	Increment Increment Word
DEC DECW	dst dst	Decrement Decrement Word
SLA SLAW	dst dst	Shift Left Arithmetic Shift Word Left Arithmetic
SRA SRAW	dst dst	Shift Right Arithmetic Shift Word Right Arithmetic
RRC RRCW	dst dst	Rotate Right Through Carry Rotate Word Right Through Carry
RLC RLCW	dst dst	Rotate Left Through Carry Rotate Word Left Through Carry
ROR	dst	Rotate Right
ROL	dst	Rotate Left
CLR	dst	Clear Register
CPL	dst	Complement Register
SWAP	dst	Swap Nibbles
DA	dst	Decimal Adjust

Stack Instructions (One Operand)

Mnemonic	Operands	Instruction
PUSH PUSHW PEA	src src src	Push on System Stack Push Word on System Stack Push Effective Address on System Stack
POP POPW	dst dst	Pop From System Stack Pop Word from System Stack
PUSHU PUSHUW PEAU	src src src	Push on User Stack Push Word on User Stack Push Effective Address on User Stack
POPU POPUW	dst dst	Pop From User Stack Pop Word From User Stack

INSTRUCTION SET (Continued)

Multiply and Divide Instructions (Two Operands)

Mnemonic	Operands	Instruction
MUL	dst,src	Multiply 8x8
DIV DIVWS	dst,src dst,src	Divide 16/8 Divide Word Stepped 32/16

Boolean Instructions (One or Two Operands)

Mnemonic	Operands	Instruction
BSET	dst	Bit Set
BRES	dst	Bit Reset
BCPL	dst	Bit Complement
BTSET	dst	Bit Test and Set
BLD	dst,src	Bit Load
BAND	dst,src	Bit AND
BOR	dst,src	Bit OR
BXOR	dst,src	Bit XOR

INSTRUCTION SET (Continued)

Program Control Instructions (One, Two or Three Operands)

Mnemonic	Operands	Instruction
RET		Return from Subroutine
IRET		Return from Interrupt
WFI		Stop Program Execution and Wait for the next Enabled Interrupt. If a DMA request is present, the CPU executes the DMA service routine and then automatically returns to the WFI
HALT		Stop Program Execution Until Next System Reset
JR	cc,dst	Jump Relative If Condition is Met
JP	cc,dst	Jump if Condition is Met
JP	dst	Unconditional Jump
CALL	dst	Unconditional Call
BTJF	dst,N	Bit Test and Jump if False
BTJT	dst,N	Bit Test and Jump if True
DJNZ	dst,N	Decrement a Working Register and Jump if Non Zero
DWJNZ	dst,N	Decrement a Register Pair and Jump if Non Zero
CPJFI	dst,N	Compare and Jump on False. Otherwise Post Increment
CPJTI	dst,N	Compare and Jump on True. Otherwise Post Increment

INSTRUCTION SET (Continued)

Miscellaneous (None, One or Two Operands)

Mnemonic	Operands	Instruction
XCH	dst,src	Exchange Registers
SRP	src	Set Register Pointer Long (16 working registers)
SRP0	src	Set Register Pointer 0 (8 LSB working register)
SRP1	src	Set Register Pointer 1 (8 MSB working register)
SPP	src	Set Page Pointer
EXT	dst	Sign Extend
EI		Enable Interrupts
DI		Disable Interrupts
SCF		Set Carry Flag
RCF		Reset Carry Flag
CCF		Complement Carry Flag
SPM		Select Program Memory
SDM		Select Data Memory
NOP		No Operation

INSTRUCTION SET (Continued)

1.2.1 ST9 Processor Flags

An important feature of a single chip microcomputer is the ability to test data and make the appropriate action based on the results. In order to provide this facility, FLAGR (register 231) in the register file is used as a flag register. Six bits of this register are used as the following flags:

C - Carry

Z - Zero

S - Sign

V - Overflow

D - Decimal Adjust

H - Half Carry

Bit 1 is available to the user. Bit 0 is the Program/Data Memory selector bit.

The Flag Register is further described in the Architecture Chapter.

1.2.2 Condition Codes

Flags C, Z, S, and OV control the operation of the "conditional" Jump instructions. The next table shows the condition codes and the flag settings.

Note : Some of the Status flags are used to indicate more than one condition e.g . Zero and Equal. In such cases the condition code is the same for both conditions.

Table 1-2. Condition Codes Table

Mnemonic code	Meaning	Flag setting	Hex. value	Binary value
F	Always False	____	0	0000
T	Always True	____	8	1000
C	Carry	C=1	7	0111
NC	Not carry	C=0	F	1111
Z	Zero	Z=1	6	0011
NZ	Not Zero	Z=0	E	1110
PL	Plus	S=0	D	1101
MI	Minus	S=1	5	0101
OV	Overflow	V=1	4	0100
NOV	No Overflow	V=0	C	1100
EQ	Equal	Z=1	6	0110
NE	Not Equal	Z=0	E	1110
GE	Greater Than or Equal	(S xor V)=0	9	1001
LT	Less Than	(S xor V)=1	1	0001
GT	Greater Than	(Z or(S xor V))=0	A	1010
LE	Less Than or Equal	(Z or(S xor V))=1	2	0010
UG	Unsigned Greater Than or Equal	C=0	F	1111
UL	Unsigned Less Than	C=1	7	0111
UGT	Unsigned Greater Than	(C=0 and Z=0)=1	B	1011
ULE	Unsigned Less Than or Equal	(C or Z)=1	3	0011

INSTRUCTION SET (Continued)

1.2.3 Notation

Operands and status flags are represented by a notational shorthand in the detailed instruction description (see programming manual).

The notation for operands (condition codes and address modes) and the actual operands they represent are as follows:

Table 1-3. Notation (Part 1)

Notation	Significance	Actual Operand/Range	
cc	Condition Code		
#N #NN	Immediate Byte Immediate Word	# data # data	where data is a byte expression where data is a word expression
r	Direct Working Register	rn	where n=0-15
R	Direct Register	Rn	where n=0-255
rr	Direct Working Register Pair	rrn	where n is an even number in the range 0-15. (n=0,2,4,6....14)
RR	Direct Register Pair	RRn	where n is an even number in the range 0-254. (n=0,2,4,6....254)
(r)	Indirect Working Register	(rn)	where n=0-15
(R)	Indirect register	(Rn)	where n=0-255
(r)+	Indirect working register post increment	(rn)+	where n=0-15
N(rx)	Indexed register	N(rx)	where x=0-15; N=0-255 (one byte)
N	Memory relative Short Address		Program label or expression in the range +127/-128 starting from the address of the next instruction
NN	Direct Memory Long Address		Program label or expression in the range 0-65535 in memory area
(rr)	Indirect Pair of Working Register Pointers	(rrn)	Where n is an even number in the range 0-15.(n=0,2,4,6....14)
(rr)+	Indirect Pair of Working Register Pointers with Post Increment	(rrn)+	where n is an even number in the range 0-15.(n=0,2,4,6....14)
-(rr)	Indirect Pair of Working Register Pointers with Pre Decrement	-(rrn)	where n is an even number in the range 0-15.(n=0,2,4,6....14)

INSTRUCTION SET (Continued)

Table 1-4. Notation (Part 2)

Notation	Significance	Actual Operand/Range	
N(rrx)	Indexed Pair of Working Register Pointers with Short Offset	N(rrx)	where x is an even number in the range 0-15.(n=0,2,4,6....14) and N is a signed one byte expression between +127/-128
NN(rrx)	Indexed Pair of Working Register Pointers with Long Offset	NN(rrx)	where x is an even number in the range 0-15.(n=0,2,4,6....14) and NN is word expression in the range between 0 and 65535
N(RRx)	Indexed Pair of Register Pointers with Short Offset	N(RRx)	where x is an even number in the range 0-255.(n=0,2,4,6...254) and N is a one byte signed expression in the range +127/-128
NN(RRx)	Indexed Pair of Register Pointers with Long Offset	NN(RRx)	where x is an even number in the range 0-255.(n=0,2,4,6....14) and NN is word expression in the range between 0 and 65535
rr(rrx)	Indexed Pair of Working Registers with a Pair of Working Registers used as Offset	rrn(rrx)	where n and x are two even numbers in the range 0-15. (n,x=0,2,4,6....14)
r.b	Bit pointer in a direct working register	rn.b	n=0.15 and b is a number between 0-7;0 LSB 7 MSB
(rr).b	Bit pointer in a Memory Location using a Pair of Indirect Working Registers as Address Pointer	(rrn).b	where n is an even number in the range 0-15.(n=0,2,4,6....14) and b is a number between 0-7 0 LSB 7 MSB
(RR)	Indirect pair of Register Pointer	(RRn)	where n is an even number in the range 0-255.(n=0,2,4,6....254)

1.3 INSTRUCTION SUMMARY

The following tables summarize the operation for each of the instructions which are listed with their corresponding mnemonic codes, addressing modes, byte counts, timing information, and affected flags.

GENERAL NOTES:

FLAGS STATUS:

- ^ : affected
- - : not affected
- 0 : reset to zero
- 1 : set to one
- ? : undefined

Note: for detailed information on the instruction set refer to the ST9 programming manual.

- dst: destination operand
- src: source operand
- SSP: system stack pointer
- USP: user stack pointer
- PC: program counter
- cc: condition code
- C: carry flag
- Z: zero flag
- S: sign flag
- V: overflow flag
- D: decimal adjust flag
- CIC: central interrupt control register
- DP : data/program memory flag

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
ADC : Addition of 2 bytes with carry						
ADC	r	r	2	6	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	R	R	3	10	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	r	R	3	10	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	R	r	3	10	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	r	(r)	2	6	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	R	(r)	3	10	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	r	(rr)	3	12	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	R	(rr)	3	12	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	r	NN	4	18	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	r	N(rrx)	4	24	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	R	N(rrx)	4	24	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	r	NN(rrx)	5	26	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	R	NN(rrx)	5	26	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	r	rr(rrx)	3	22	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	r	(rr)+	3	16	dst<-dst+src+C rr<-rr+1	^ ^ ^ ^ 0 ^
ADC	R	(rr)+	3	16	dst<-dst+src+C rr<-rr+1	^ ^ ^ ^ 0 ^
ADC	r	-(rr)	3	16	rr<-rr-1 dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	R	-(rr)	3	16	rr<-rr-1 dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	(r)	r	3	10	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	(r)	R	3	10	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	(rr)	r	3	18	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	(rr)	R	3	18	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	(rr)+	r	3	22	dst<-dst+src+C rr<-rr+1	^ ^ ^ ^ 0 ^
ADC	(rr)+	R	3	22	dst<-dst+src+C rr<-rr+1	^ ^ ^ ^ 0 ^
ADC	NN	r	4	20	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	N(rrx)	r	4	26	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	N(rrx)	R	4	26	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	NN(rrx)	r	5	28	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	NN(rrx)	R	5	28	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	rr(rrx)	r	3	24	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	-(rr)	r	3	24	rr<-rr-1 dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	-(rr)	R	3	22	rr<-rr-1 dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	r	#N	3	10	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	R	#N	3	10	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	(rr)	#N	3	16	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	NN	#N	5	24	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	(rr)	(rr)	3	20	dst<-dst+src+C	^ ^ ^ ^ 0 ^
ADC	(RR)	(rr)	3	20	dst<-dst+src+C	^ ^ ^ ^ 0 ^

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
ADCW : Add word with carry						
ADCW	rr	rr	2	10	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	RR	RR	3	12	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	rr	RR	3	12	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	RR	rr	3	12	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	rr	(r)	3	14	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	RR	(r)	3	14	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	rr	(rr)	2	16	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	RR	(rr)	3	18	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	rr	NN	4	22	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	rr	N(rrx)	4	28	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	RR	N(rrx)	4	28	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	rr	NN(rrx)	5	30	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	RR	NN(rrx)	5	30	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	rr	rr(rrx)	3	26	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	rr	(rr)+	3	22	dst<-dst+src+C rr<-rr+2	^ ^ ^ ^ ? ?
ADCW	RR	(rr)+	3	22	dst<-dst+src+C rr<-rr+2	^ ^ ^ ^ ? ?
ADCW	rr	-(rr)	3	24	rr<-rr-2 dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	RR	-(rr)	3	24	rr<-rr-2 dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	(r)	rr	3	14	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	(r)	RR	3	14	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	(rr)	rr	2	30	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	(rr)	RR	3	30	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	(rr)+	rr	3	32	dst<-dst+src+C rr<-rr+2	^ ^ ^ ^ ? ?
ADCW	(rr)+	RR	3	32	dst<-dst+src+C rr<-rr+2	^ ^ ^ ^ ? ?
ADCW	NN	rr	4	32	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	N(rrx)	rr	4	38	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	N(rrx)	RR	4	38	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	NN(rrx)	rr	5	38	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	NN(rrx)	RR	5	38	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	rr(rrx)	rr	3	34	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	-(rr)	rr	3	34	rr<-rr-2 dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	-(rr)	RR	3	32	rr<-rr-2 dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	rr	#NN	4	14	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	RR	#NN	4	14	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	(rr)	#NN	4	32	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	NN	#NN	6	36	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	N(rrx)	#NN	5	36	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	NN(rrx)	#NN	6	38	dst<-dst+src+C	^ ^ ^ ^ ? ?
ADCW	(rr)	(rr)	2	32	dst<-dst+src+C	^ ^ ^ ^ ? ?

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
ADD : Addition of 2 bytes without carry						
ADD	r	r	2	6	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	R	R	3	10	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	r	R	3	10	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	R	r	3	10	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	r	(r)	2	6	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	R	(r)	3	10	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	r	(rr)	3	12	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	R	(rr)	3	12	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	r	NN	4	18	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	r	N(rrx)	4	24	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	R	N(rrx)	4	24	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	r	NN(rrx)	5	26	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	R	NN(rrx)	5	26	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	r	rr(rrx)	3	22	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	r	(rr)+	3	16	dst<-dst+src rr<-rr+1	^ ^ ^ ^ 0 ^
ADD	R	(rr)+	3	16	dst<-dst+src rr<-rr+1	^ ^ ^ ^ 0 ^
ADD	r	-(rr)	3	16	rr<-rr-1 dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	R	-(rr)	3	16	rr<-rr-1 dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	(r)	r	3	10	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	(r)	R	3	10	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	(rr)	r	3	18	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	(rr)	R	3	18	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	(rr)+	r	3	22	dst<-dst+src rr<-rr+1	^ ^ ^ ^ 0 ^
ADD	(rr)+	R	3	22	dst<-dst+src rr<-rr+1	^ ^ ^ ^ 0 ^
ADD	NN	r	4	20	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	N(rrx)	r	4	26	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	N(rrx)	R	4	26	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	NN(rrx)	r	5	28	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	NN(rrx)	R	5	28	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	rr(rrx)	r	3	24	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	-(rr)	r	3	22	rr<-rr-1 dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	-(rr)	R	3	22	rr<-rr-1 dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	r	#N	3	10	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	R	#N	3	10	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	(rr)	#N	3	16	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	NN	#N	5	24	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	(rr)	(rr)	3	20	dst<-dst+src	^ ^ ^ ^ 0 ^
ADD	(RR)	(rr)	3	20	dst<-dst+src	^ ^ ^ ^ 0 ^

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
ADDW : Add word without carry						
ADDW	rr	rr	2	10	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	RR	RR	3	12	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	rr	RR	3	12	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	RR	rr	3	12	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	rr	(r)	3	14	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	RR	(r)	3	14	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	rr	(rr)	2	16	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	RR	(rr)	3	18	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	rr	NN	4	22	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	rr	N(rrx)	4	28	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	RR	N(rrx)	4	28	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	rr	NN(rrx)	5	30	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	RR	NN(rrx)	5	30	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	rr	rr(rrx)	3	26	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	rr	(rr)+	3	22	dst<-dst+src rr<-rr+2	^ ^ ^ ^ ? ?
ADDW	RR	(rr)+	3	22	dst<-dst+src rr<-rr+2	^ ^ ^ ^ ? ?
ADDW	rr	-(rr)	3	24	rr<-rr-2 dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	RR	-(rr)	3	24	rr<-rr-2 dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	(r)	rr	3	14	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	(r)	RR	3	14	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	(rr)	rr	2	30	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	(rr)	RR	3	30	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	(rr)+	rr	3	32	dst<-dst+src rr<-rr+2	^ ^ ^ ^ ? ?
ADDW	(rr)+	RR	3	32	dst<-dst+src rr<-rr+2	^ ^ ^ ^ ? ?
ADDW	NN	rr	4	32	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	N(rrx)	rr	4	38	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	N(rrx)	RR	4	38	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	NN(rrx)	rr	5	38	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	NN(rrx)	RR	5	38	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	rr(rrx)	rr	3	34	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	-(rr)	rr	3	32	rr<-rr-2 dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	-(rr)	RR	3	32	rr<-rr-2 dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	rr	#NN	4	14	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	RR	#NN	4	14	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	(rr)	#NN	4	32	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	NN	#NN	6	36	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	N(rrx)	#NN	5	36	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	NN(rrx)	#NN	6	38	dst<-dst+src	^ ^ ^ ^ ? ?
ADDW	(rr)	(rr)	2	32	dst<-dst+src	^ ^ ^ ^ ? ?

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
AND : Logical AND between 2 bytes						
AND	r	r	2	6	dst<-dst AND src	- ^ ^ 0 - -
AND	R	R	3	10	dst<-dst AND src	- ^ ^ 0 - -
AND	r	R	3	10	dst<-dst AND src	- ^ ^ 0 - -
AND	R	r	3	10	dst<-dst AND src	- ^ ^ 0 - -
AND	r	(r)	2	6	dst<-dst AND src	- ^ ^ 0 - -
AND	R	(r)	3	10	dst<-dst AND src	- ^ ^ 0 - -
AND	r	(rr)	3	12	dst<-dst AND src	- ^ ^ 0 - -
AND	R	(rr)	3	12	dst<-dst AND src	- ^ ^ 0 - -
AND	r	NN	4	18	dst<-dst AND src	- ^ ^ 0 - -
AND	r	N(rrx)	4	24	dst<-dst AND src	- ^ ^ 0 - -
AND	R	N(rrx)	4	24	dst<-dst AND src	- ^ ^ 0 - -
AND	r	NN(rrx)	5	26	dst<-dst AND src	- ^ ^ 0 - -
AND	R	NN(rrx)	5	26	dst<-dst AND src	- ^ ^ 0 - -
AND	r	rr(rrx)	3	22	dst<-dst AND src	- ^ ^ 0 - -
AND	r	(rr)+	3	16	dst<-dst AND src rr<-rr+1	- ^ ^ 0 - -
AND	R	(rr)+	3	16	dst<-dst AND src rr<-rr+1	- ^ ^ 0 - -
AND	r	-(rr)	3	16	rr<-rr-1 dst<-dst AND src	- ^ ^ 0 - -
AND	R	-(rr)	3	16	rr<-rr-1 dst<-dst AND src	- ^ ^ 0 - -
AND	(r)	r	3	10	dst<-dst AND src	- ^ ^ 0 - -
AND	(r)	R	3	10	dst<-dst AND src	- ^ ^ 0 - -
AND	(rr)	r	3	18	dst<-dst AND src	- ^ ^ 0 - -
AND	(rr)	R	3	18	dst<-dst AND src	- ^ ^ 0 - -
AND	(rr)+	r	3	22	dst<-dst AND src rr<-rr+1	- ^ ^ 0 - -
AND	(rr)+	R	3	22	dst<-dst AND src rr<-rr+1	- ^ ^ 0 - -
AND	NN	r	4	20	dst<-dst AND src	- ^ ^ 0 - -
AND	N(rrx)	r	4	26	dst<-dst AND src	- ^ ^ 0 - -
AND	N(rrx)	R	4	26	dst<-dst AND src	- ^ ^ 0 - -
AND	NN(rrx)	r	5	28	dst<-dst AND src	- ^ ^ 0 - -
AND	NN(rrx)	R	5	28	dst<-dst AND src	- ^ ^ 0 - -
AND	rr(rrx)	r	3	24	dst<-dst AND src	- ^ ^ 0 - -
AND	-(rr)	r	3	22	rr<-rr-1 dst<-dst AND src	- ^ ^ 0 - -
AND	-(rr)	R	3	22	rr<-rr-1 dst<-dst AND src	- ^ ^ 0 - -
AND	r	#N	3	10	dst<-dst AND src	- ^ ^ 0 - -
AND	R	#N	3	10	dst<-dst AND src	- ^ ^ 0 - -
AND	(rr)	#N	3	16	dst<-dst AND src	- ^ ^ 0 - -
AND	NN	#N	5	24	dst<-dst AND src	- ^ ^ 0 - -
AND	(rr)	(rr)	3	20	dst<-dst AND src	- ^ ^ 0 - -
AND	(RR)	(rr)	3	20	dst<-dst AND src	- ^ ^ 0 - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
ANDW : Logical AND between two words						
ANDW	rr	rr	2	10	dst<-dst AND src	- ^ ^ 0 - -
ANDW	RR	RR	3	12	dst<-dst AND src	- ^ ^ 0 - -
ANDW	rr	RR	3	12	dst<-dst AND src	- ^ ^ 0 - -
ANDW	RR	rr	3	12	dst<-dst AND src	- ^ ^ 0 - -
ANDW	rr	(r)	3	14	dst<-dst AND src	- ^ ^ 0 - -
ANDW	RR	(r)	3	14	dst<-dst AND src	- ^ ^ 0 - -
ANDW	rr	(rr)	2	16	dst<-dst AND src	- ^ ^ 0 - -
ANDW	RR	(rr)	3	18	dst<-dst AND src	- ^ ^ 0 - -
ANDW	rr	NN	4	22	dst<-dst AND src	- ^ ^ 0 - -
ANDW	rr	N(rrx)	4	28	dst<-dst AND src	- ^ ^ 0 - -
ANDW	RR	N(rrx)	4	28	dst<-dst AND src	- ^ ^ 0 - -
ANDW	rr	NN(rrx)	5	30	dst<-dst AND src	- ^ ^ 0 - -
ANDW	RR	NN(rrx)	5	30	dst<-dst AND src	- ^ ^ 0 - -
ANDW	rr	rr(rrx)	3	26	dst<-dst AND src	- ^ ^ 0 - -
ANDW	rr	(rr)+	3	22	dst<-dst AND src rr<-rr+2	- ^ ^ 0 - -
ANDW	RR	(rr)+	3	22	dst<-dst AND src rr<-rr+2	- ^ ^ 0 - -
ANDW	rr	-(rr)	3	24	rr<-rr-2 dst<-dst AND src	- ^ ^ 0 - -
ANDW	RR	-(rr)	3	24	rr<-rr-2 dst<-dst AND src	- ^ ^ 0 - -
ANDW	(r)	rr	3	14	dst<-dst AND src	- ^ ^ 0 - -
ANDW	(r)	RR	3	14	dst<-dst AND src	- ^ ^ 0 - -
ANDW	(rr)	rr	2	30	dst<-dst AND src	- ^ ^ 0 - -
ANDW	(rr)	RR	3	30	dst<-dst AND src	- ^ ^ 0 - -
ANDW	(rr)+	rr	3	32	dst<-dst AND src rr<-rr+2	- ^ ^ 0 - -
ANDW	(rr)+	RR	3	32	dst<-dst AND src rr<-rr+2	- ^ ^ 0 - -
ANDW	NN	rr	4	32	dst<-dst AND src	- ^ ^ 0 - -
ANDW	N(rrx)	rr	4	38	dst<-dst AND src	- ^ ^ 0 - -
ANDW	N(rrx)	RR	4	38	dst<-dst AND src	- ^ ^ 0 - -
ANDW	NN(rrx)	rr	5	38	dst<-dst AND src	- ^ ^ 0 - -
ANDW	NN(rrx)	RR	5	38	dst<-dst AND src	- ^ ^ 0 - -
ANDW	rr(rrx)	rr	3	34	dst<-dst AND src	- ^ ^ 0 - -
ANDW	-(rr)	rr	3	32	rr<-rr-2 dst<-dst AND src	- ^ ^ 0 - -
ANDW	-(rr)	RR	3	32	rr<-rr-2 dst<-dst AND src	- ^ ^ 0 - -
ANDW	rr	#NN	4	14	dst<-dst AND src	- ^ ^ 0 - -
ANDW	RR	#NN	4	14	dst<-dst AND src	- ^ ^ 0 - -
ANDW	(rr)	#NN	4	32	dst<-dst AND src	- ^ ^ 0 - -
ANDW	NN	#NN	6	36	dst<-dst AND src	- ^ ^ 0 - -
ANDW	N(rrx)	#NN	5	36	dst<-dst AND src	- ^ ^ 0 - -
ANDW	NN(rrx)	#NN	6	38	dst<-dst AND src	- ^ ^ 0 - -
ANDW	(rr)	(rr)	2	32	dst<-dst AND src	- ^ ^ 0 - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
BAND : Bit AND						
BAND BAND	r.b r.b	r.b r.!b	3 3	14 14	dst bit<-dst bit AND src bit dst bit<-dst bit AND complemented src bit	- - - - - - - - - - - -
BCPL : Bit Complement						
BCPL	r.b		2	6	dst bit<-dst bit complemented	- - - - - -
BLD : Bit Load						
BLD BLD	r.b r.b	r.b r.!b	3 3	14 14	dst bit<-src bit dst bit<-src bit complemented	- - - - - - - - - - - -
BOR : Bit OR						
BOR BOR	r.b r.b	r.b r.!b	3 3	14 14	dst bit<-dst bit OR src bit dst bit<-dst bit OR complemented src bit	- - - - - - - - - - - -
BRES : Bit Reset						
BRES	r.b		2	6	dst bit<- 0	- - - - - -
BSET : Bit Set						
BSET	r.b		2	6	dst bit<- 1	- - - - - -
BTJF, BTJT : Bit test and jump						
BTJF	r.b	N	3	14/16	If test bit is 0, PC<-PC+N	- - - - - -
BTJT	r.b	N	3	14/16	If test bit is 1, PC<-PC+N	- - - - - -
BXOR : Bit Exclusive OR						
BXOR BXOR	r.b r.b	r.b r.!b	3 3	14 14	dst bit<-dst bit XOR src bit dst bit<-dst bit XOR complemented src bit	- - - - - - - - - - - -
BTSET : Bit Test and Set						
BTSET BTSET	r.b (rr).b		2 2	8 20	If test bit = 0, test bit <-1,Z<-1 If test bit = 0, test bit <-1,Z<-1	- ^ ^ 0 - - - ^ ^ 0 - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
CALL : Call a subroutine						
CALL	NN		3	18	SSP<-SSP-2,(SP)< PC, PC<-dst	- - - - - -
CALL	(rr)		2	16	" "	- - - - - -
CALL	(RR)		2	16	" "	- - - - - -
CCF : Complement Carry Flag						
CCF			1	6	C <- C complemented	- - - - - -
CLR : Clear register						
CLR	r		2	6	dst<-0	- - - - - -
CLR	R		2	6	dst<-0	- - - - - -
CLR	(r)		2	6	dst<-0	- - - - - -
CLR	(R)		2	6	dst<-0	- - - - - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
CP : Compare bytes						
CP	r	r	2	6	dst-src	^ ^ ^ ^ - -
CP	R	R	3	10	dst-src	^ ^ ^ ^ - -
CP	r	R	3	10	dst-src	^ ^ ^ ^ - -
CP	R	r	3	10	dst-src	^ ^ ^ ^ - -
CP	r	(r)	2	6	dst-src	^ ^ ^ ^ - -
CP	R	(r)	3	10	dst-src	^ ^ ^ ^ - -
CP	r	(rr)	3	12	dst-src	^ ^ ^ ^ - -
CP	R	(rr)	3	12	dst-src	^ ^ ^ ^ - -
CP	r	NN	4	18	dst-src	^ ^ ^ ^ - -
CP	r	N(rrx)	4	24	dst-src	^ ^ ^ ^ - -
CP	R	N(rrx)	4	24	dst-src	^ ^ ^ ^ - -
CP	r	NN(rrx)	5	26	dst-src	^ ^ ^ ^ - -
CP	R	NN(rrx)	5	26	dst-src	^ ^ ^ ^ - -
CP	r	rr(rrx)	3	22	dst-src	^ ^ ^ ^ - -
CP	r	(rr)+	3	16	dst-src,rr<-rr+1	^ ^ ^ ^ - -
CP	R	(rr)+	3	16	dst-src,rr<-rr+1	^ ^ ^ ^ - -
CP	r	-(rr)	3	16	rr<-rr-1,dst-src	^ ^ ^ ^ - -
CP	R	-(rr)	3	16	rr<-rr-1,dst-src	^ ^ ^ ^ - -
CP	(r)	r	3	10	dst-src	^ ^ ^ ^ - -
CP	(r)	R	3	10	dst-src	^ ^ ^ ^ - -
CP	(rr)	r	3	18	dst-src	^ ^ ^ ^ - -
CP	(rr)	R	3	18	dst-src	^ ^ ^ ^ - -
CP	(rr)+	r	3	22	dst-src,rr<-rr+1	^ ^ ^ ^ - -
CP	(rr)+	R	3	22	dst-src,rr<-rr+1	^ ^ ^ ^ - -
CP	NN	r	4	20	dst-src	^ ^ ^ ^ - -
CP	N(rrx)	r	4	26	dst-src	^ ^ ^ ^ - -
CP	N(rrx)	R	4	26	dst-src	^ ^ ^ ^ - -
CP	NN(rrx)	r	5	28	dst-src	^ ^ ^ ^ - -
CP	NN(rrx)	R	5	28	dst-src	^ ^ ^ ^ - -
CP	rr(rrx)	r	3	24	dst-src	^ ^ ^ ^ - -
CP	-(rr)	r	3	22	rr<-rr-1,dst-src	^ ^ ^ ^ - -
CP	-(rr)	R	3	22	rr<-rr-1,dst-src	^ ^ ^ ^ - -
CP	r	#N	3	10	dst-src	^ ^ ^ ^ - -
CP	R	#N	3	10	dst-src	^ ^ ^ ^ - -
CP	(rr)	#N	3	16	dst-src	^ ^ ^ ^ - -
CP	NN	#N	5	22	dst-src	^ ^ ^ ^ - -
CP	(rr)	(rr)	3	18	dst-src	^ ^ ^ ^ - -
CP	(RR)	(rr)	3	18	dst-src	^ ^ ^ ^ - -
CPL : Complement register						
CPL	r		2	6	dst<- NOT dst	- ^ ^ 0 - -
CPL	R		2	6	dst<- NOT dst	- ^ ^ 0 - -
CPL	(r)		2	6	dst<- NOT dst	- ^ ^ 0 - -
CPL	(R)		2	6	dst<- NOT dst	- ^ ^ 0 - -
CPJFI, CPJTI : Compare with post-increment						
CPJFI	(rr)	r,N	3	22/24	If compare not verified jump otherwise post-increment	- - - - - -
CPJTI	(rr)	r,N	3	22/24	If compare verified jump otherwise post-increment	- - - - - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
CPW : Compare word						
CPW	rr	rr	2	10	dst-src	^ ^ ^ ^ - -
CPW	RR	RR	3	12	dst-src	^ ^ ^ ^ - -
CPW	rr	RR	3	12	dst-src	^ ^ ^ ^ - -
CPW	RR	rr	3	12	dst-src	^ ^ ^ ^ - -
CPW	rr	(r)	3	14	dst-src	^ ^ ^ ^ - -
CPW	RR	(r)	3	14	dst-src	^ ^ ^ ^ - -
CPW	rr	(rr)	2	16	dst-src	^ ^ ^ ^ - -
CPW	RR	(rr)	3	18	dst-src	^ ^ ^ ^ - -
CPW	rr	NN	4	22	dst-src	^ ^ ^ ^ - -
CPW	rr	N(rrx)	4	28	dst-src	^ ^ ^ ^ - -
CPW	RR	N(rrx)	4	28	dst-src	^ ^ ^ ^ - -
CPW	rr	NN(rrx)	5	30	dst-src	^ ^ ^ ^ - -
CPW	RR	NN(rrx)	5	30	dst-src	^ ^ ^ ^ - -
CPW	rr	rr(rrx)	3	26	dst-src	^ ^ ^ ^ - -
CPW	rr	(rr)+	3	22	dst-src rr<-rr+2	^ ^ ^ ^ - -
CPW	RR	(rr)+	3	22	dst-src rr<-rr+2	^ ^ ^ ^ - -
CPW	rr	-(rr)	3	24	rr<-rr-2 dst-src	^ ^ ^ ^ - -
CPW	RR	-(rr)	3	24	rr<-rr-2 dst-src	^ ^ ^ ^ - -
CPW	(r)	rr	3	14	dst-src	^ ^ ^ ^ - -
CPW	(r)	RR	3	14	dst-src	^ ^ ^ ^ - -
CPW	(rr)	rr	2	26	dst-src	^ ^ ^ ^ - -
CPW	(rr)	RR	3	28	dst-src	^ ^ ^ ^ - -
CPW	(rr)+	rr	3	30	dst-src rr<-rr+2	^ ^ ^ ^ - -
CPW	(rr)+	RR	3	30	dst-src rr<-rr+2	^ ^ ^ ^ - -
CPW	NN	rr	4	30	dst-src	^ ^ ^ ^ - -
CPW	N(rrx)	rr	4	36	dst-src	^ ^ ^ ^ - -
CPW	N(rrx)	RR	4	36	dst-src	^ ^ ^ ^ - -
CPW	NN(rrx)	rr	5	36	dst-src	^ ^ ^ ^ - -
CPW	NN(rrx)	RR	5	36	dst-src	^ ^ ^ ^ - -
CPW	rr(rrx)	rr	3	32	dst-src	^ ^ ^ ^ - -
CPW	-(rr)	rr	3	30	rr<-rr-2 dst-src	^ ^ ^ ^ - -
CPW	-(rr)	RR	3	30	rr<-rr-2 dst-src	^ ^ ^ ^ - -
CPW	rr	#NN	4	14	dst-src	^ ^ ^ ^ - -
CPW	RR	#NN	4	14	dst-src	^ ^ ^ ^ - -
CPW	(rr)	#NN	4	30	dst-src	^ ^ ^ ^ - -
CPW	NN	#NN	6	34	dst-src	^ ^ ^ ^ - -
CPW	N(rrx)	#NN	5	34	dst-src	^ ^ ^ ^ - -
CPW	NN(rrx)	#NN	6	36	dst-src	^ ^ ^ ^ - -
CPW	(rr)	(rr)	2	32	dst-src	^ ^ ^ ^ - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
DA : Decimal adjust						
DA	r		2	6	dst<- DA dst	^ ^ ^ ? - -
DA	R		2	6	dst<- DA dst	^ ^ ^ ? - -
DA	(r)		2	6	dst<- DA dst	^ ^ ^ ? - -
DA	(R)		2	6	dst<- DA dst	^ ^ ^ ? - -
DEC : Decrement						
DEC	r		2	6	dst<- dst-1	- ^ ^ ^ - -
DEC	R		2	6	dst<- dst-1	- ^ ^ ^ - -
DEC	(r)		2	6	dst<- dst-1	- ^ ^ ^ - -
DEC	(R)		2	6	dst<- dst-1	- ^ ^ ^ - -
DECW : Decrement Word						
DECW	rr		2	8	dst<-dst-1	- ^ ^ ^ - -
DECW	RR		2	8	dst<-dst-1	- ^ ^ ^ - -
DI : Disable Interrupts						
DI			1	6	Bit 4 of the CIC Register is set to 0	- - - - - -
DIV : Divide 16 by 8						
DIV	rr	r	2	28/20	dst / src <- dst high=remainder 16/8 <- dst low=result	note 1
DIVWS : Divide Word Stepped 32 by 16						
DIVWS	rrhigh rrlow	rr	3	28	32/16	note 1
DJNZ : Decrement a working register and Jump if Non Zero						
DJNZ	r	N	2	10/12	r <- r-1, If r=0 then PC<-PC+N	note 2
DWJNZ : Decrement a register pair and Jump if Non Zero						
DWJNZ	rr	N	3	12/16	rr<-rr-1, If rr=0 then PC<-PC+N	note 2
DWJNZ	RR	N	3	12/16	RR<-RR-1,If RR=0 then PC<-PC+N	

Notes :
1. Refer to the ST9 Programming Manual for detailed information.
2. Working registers in groups D, E and F are not allowed.

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
EI : Enable Interrupts						
EI			1	6	Bit 4 of the CICR register is set to 1	- - - - - -
EXT : Sign extend						
EXT EXT	rr RR		2 2	10 10	r(7) —> r(n) n=8-15 R(7) —> R(n) n=8-15	- - - - - - - - - - - -
HALT : Halt Operation						
HALT			2	6	Stops all internal clocks until next system reset if not in Watchdog Mode	- - - - - -
INC : Increment						
INC INC INC INC	r R (r) (R)		2 2 2 2	6 6 6 6	dst<- dst+1 dst<- dst+1 dst<- dst+1 dst<- dst+1	- ^ ^ ^ - - - ^ ^ ^ - - - ^ ^ ^ - - - ^ ^ ^ - -
INCW : Increment Word						
INCW INCW	rr RR		2 2	8 8	dst<-dst+1 dst<-dst+1	- ^ ^ ^ - - - ^ ^ ^ - -
IRET : Return from Interrupt Routine						
IRET			1	16	FLAGS<-(SSP),SSP<-SSP+1, PC<-(SSP), SSP<-SPP+2, CIC(4)<-1	note 1
JP : Jump to a Routine						
JP	NN		3	10	PC<-dst	- - - - - -
JP	(rr)		2	8	PC<-dst	- - - - - -
JP	(RR)		2	8	PC<-dst	- - - - - -
JPcc	NN		3	10	IF cc(condition code) is true, PC<-dst	- - - - - -
JRcc : Conditional Relative Jump to a Routine						
JRcc	N		2	10/12	IF cc(condition code)is true, PC<-PC+dst	- - - - - -

Note 1 : All flags are restored to original setting (before interrupt occured).

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
LD : Load byte instructions						
LD	r	r	2	6	dst<-src	- - - - - -
LD	R	R	3	10	dst<-src	- - - - - -
LD	r	R	2	6	dst<-src	- - - - - -
LD	R	r	2	6	dst<-src	- - - - - -
LD	r	(r)	2	6	dst<-src	- - - - - -
LD	R	(r)	3	10	dst<-src	- - - - - -
LD	r	(rr)	2	10	dst<-src	- - - - - -
LD	R	(rr)	3	12	dst<-src	- - - - - -
LD	r	NN	4	18	dst<-src	- - - - - -
LD	r	N(rx)	3	10	dst<-src	- - - - - -
LD	r	N(rrx)	4	24	dst<-src	- - - - - -
LD	R	N(rrx)	4	24	dst<-src	- - - - - -
LD	r	NN(rrx)	5	26	dst<-src	- - - - - -
LD	R	NN(rrx)	5	26	dst<-src	- - - - - -
LD	r	rr(rrx)	3	22	dst<-src	- - - - - -
LD	r	(rr)+	3	16	dst<-src,rr<-rr+1	- - - - - -
LD	R	(rr)+	3	16	dst<-src,rr<-rr+1	- - - - - -
LD	r	-(rr)	3	16	rr<-rr-1,dst<-src	- - - - - -
LD	R	-(rr)	3	16	rr<-rr-1,dst<-src	- - - - - -
LD	(r)	r	2	6	dst<-src	- - - - - -
LD	(r)	R	3	10	dst<-src	- - - - - -
LD	(rr)	r	2	10	dst<-src	- - - - - -
LD	(rr)	R	3	14	dst<-src	- - - - - -
LD	(rr)+	r	3	18	dst<-src,rr<-rr+1	- - - - - -
LD	(rr)+	R	3	18	dst<-src,rr<-rr+1	- - - - - -
LD	NN	r	4	18	dst-src	- - - - - -
LD	N(rx)	r	3	10	dst-src	- - - - - -
LD	N(rrx)	r	4	24	dst-src	- - - - - -
LD	N(rrx)	R	4	24	dst-src	- - - - - -
LD	NN(rrx)	r	5	26	dst-src	- - - - - -
LD	NN(rrx)	R	5	26	dst-src	- - - - - -
LD	rr(rrx)	r	3	22	dst-src	- - - - - -
LD	-(rr)	r	3	18	rr<-rr-1,dst<-src	- - - - - -
LD	-(rr)	R	3	18	rr<-rr-1,dst<-src	- - - - - -
LD	r	#N	2	6	dst<-src	- - - - - -
LD	R	#N	3	10	dst<-src	- - - - - -
LD	(rr)	#N	3	12	dst<-src	- - - - - -
LD	NN	#N	5	20	dst<-src	- - - - - -
LD	(rr)	(rr)	3	16	dst<-src	- - - - - -
LD	(RR)	(rr)	3	16	dst<-src,	- - - - - -
LD	(r)+	(rr)+	2	14	rr<-rr+1,r<-r+1	- - - - - -
LD	(rr)+	(r)+	2	18	rr<-rr+1,r<-r+1	- - - - - -
LDPP,LDDP,LDPD, LDDD : Load from / to program / data memory						
LDPP	(rr)+	(rr)+	2	16	dst<-src (1),rr<-rr+1	- - - - - -
LDDP	(rr)+	(rr)+	2	16	dst<-src (2),rr<-rr+1	- - - - - -
LDPD	(rr)+	(rr)+	2	16	dst<-src (3),rr<-rr+1	- - - - - -
LDDD	(rr)+	(rr)+	2	16	dst<-src (4),rr<-rr+1	- - - - - -

Notes:
1. dst in Program Memory, src in Program Memory
2. dst in Data Memory, src in Program Memory
3. dst in Program Memory, src in Data Memory
4. dst in Data Memory, src in Data Memory

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
LDW : Load word instructions						
LDW	rr	rr	2	10	dst<-src	- - - - - -
LDW	RR	RR	3	10	dst<-src	- - - - - -
LDW	rr	RR	3	10	dst<-src	- - - - - -
LDW	RR	rr	3	10	dst<-src	- - - - - -
LDW	rr	(r)	3	10	dst<-src	- - - - - -
LDW	RR	(r)	3	10	dst<-src	- - - - - -
LDW	rr	(rr)	2	16	dst<-src	- - - - - -
LDW	RR	(rr)	3	18	dst<-src	- - - - - -
LDW	rr	NN	4	22	dst<-src	- - - - - -
LDW	rr	N(rx)	3	16	dst<-src	- - - - - -
LDW	rr	N(rrx)	4	28	dst<-ssc	- - - - - -
LDW	RR	N(rrx)	4	28	dst<-src	- - - - - -
LDW	rr	NN(rrx)	5	30	dst<-src	- - - - - -
LDW	RR	NN(rrx)	5	30	dst<-src	- - - - - -
LDW	rr	rr(rrx)	3	24	dst<-src	- - - - - -
LDW	rr	(rr)+	3	20	dst<-src,rr<-rr+2	- - - - - -
LDW	RR	(rr)+	3	20	dst<-src,rr<-rr+2	- - - - - -
LDW	rr	-(rr)	3	22	rr<-rr-2,dst<-src	- - - - - -
LDW	RR	-(rr)	3	22	rr<-rr-2,dst<-src	- - - - - -
LDW	(r)	rr	3	10	dst<-src	- - - - - -
LDW	(r)	RR	3	10	dst<-src	- - - - - -
LDW	(rr)	rr	2	18	dst<-src	- - - - - -
LDW	(rr)	RR	3	20	dst<-src	- - - - - -
LDW	(rr)+	rr	3	24	rr<-rr+2,dst<-src	- - - - - -
LDW	(rr)+	RR	3	24	rr<-rr+2,dst<-src	- - - - - -
LDW	NN	rr	4	22	dst<-src	- - - - - -
LDW	N(rx)	rr	3	14	dst<-src	- - - - - -
LDW	N(rrx)	RR	4	26	dst<-src	- - - - - -
LDW	N(rrx)	rr	4	26	dst<-src	- - - - - -
LDW	NN(rrx)	RR	5	28	dst<-src	- - - - - -
LDW	NN(rrx)	rr	5	28	dst<-src	- - - - - -
LDW	rr(rrx)	rr	3	24	dst<-src	- - - - - -
LDW	-(rr)	rr	3	26	rr<-rr-2,dst<-src	- - - - - -
LDW	-(rr)	RR	3	26	rr<-rr-2,dst<-src	- - - - - -
LDW	rr	#NN	4	12	dst<-src	- - - - - -
LDW	RR	#NN	4	12	dst<-src	- - - - - -
LDW	(rr)	#NN	4	22	dst<-src	- - - - - -
LDW	N(rrx)	#NN	5	28	dst<-src	- - - - - -
LDW	NN(rrx)	#NN	6	30	dst<-src	- - - - - -
LDW	NN	#NN	6	26	dst<-src	- - - - - -
LDW	(rr)	(rr)	2	22	dst<-src	- - - - - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
MUL : Multiply						
MUL	rr	r	2	22	dst <- dst x src, 8 x 8 multiply	note 1
NOP : No operation						
NOP			1	6	No Operation	- - - - - -
OR : Logical OR between 2 bytes						
OR	r	r	2	6	dst<-dst OR src	- ^ ^ 0 - -
OR	R	R	3	10	dst<-dst OR src	- ^ ^ 0 - -
OR	r	R	3	10	dst<-dst OR src	- ^ ^ 0 - -
OR	R	r	3	10	dst<-dst OR src	- ^ ^ 0 - -
OR	r	(r)	2	6	dst<-dst OR src	- ^ ^ 0 - -
OR	R	(r)	3	10	dst<-dst OR src	- ^ ^ 0 - -
OR	r	(rr)	3	12	dst<-dst OR src	- ^ ^ 0 - -
OR	R	(rr)	3	12	dst<-dst OR src	- ^ ^ 0 - -
OR	r	NN	4	18	dst<-dst OR src	- ^ ^ 0 - -
OR	r	N(rrx)	4	24	dst<-dst OR src	- ^ ^ 0 - -
OR	R	N(rrx)	4	24	dst<-dst OR src	- ^ ^ 0 - -
OR	r	NN(rrx)	5	26	dst<-dst OR src	- ^ ^ 0 - -
OR	R	NN(rrx)	5	26	dst<-dst OR src	- ^ ^ 0 - -
OR	r	rr(rrx)	3	22	dst<-dst OR src	- ^ ^ 0 - -
OR	r	(rr)+	3	16	dst<-dst OR src rr<-rr+1	- ^ ^ 0 - -
OR	R	(rr)+	3	16	dst<-dst OR src rr<-rr+1	- ^ ^ 0 - -
OR	r	-(rr)	3	16	rr<-rr-1 dst<-dst OR src	- ^ ^ 0 - -
OR	R	-(rr)	3	16	rr<-rr-1 dst<-dst OR src	- ^ ^ 0 - -
OR	(r)	r	3	10	dst<-dst OR src	- ^ ^ 0 - -
OR	(r)	R	3	10	dst<-dst OR src	- ^ ^ 0 - -
OR	(rr)	r	3	18	dst<-dst OR src	- ^ ^ 0 - -
OR	(rr)	R	3	18	dst<-dst OR src	- ^ ^ 0 - -
OR	(rr)+	r	3	22	dst<-dst OR src rr<-rr+1	- ^ ^ 0 - -
OR	(rr)+	R	3	22	dst<-dst OR src rr<-rr+1	- ^ ^ 0 - -
OR	NN	r	4	20	dst<-dst OR src	- ^ ^ 0 - -
OR	N(rrx)	r	4	26	dst<-dst OR src	- ^ ^ 0 - -
OR	N(rrx)	R	4	26	dst<-dst OR src	- ^ ^ 0 - -
OR	NN(rrx)	r	5	28	dst<-dst OR src	- ^ ^ 0 - -
OR	NN(rrx)	R	5	28	dst<-dst OR src	- ^ ^ 0 - -
OR	rr(rrx)	r	3	24	dst<-dst OR src	- ^ ^ 0 - -
OR	-(rr)	r	3	22	dst<-dst OR src rr<-rr-1	- ^ ^ 0 - -
OR	-(rr)	R	3	22	dst<-dst OR src rr<-rr-1	- ^ ^ 0 - -
OR	r	#N	3	10	dst<-dst OR src	- ^ ^ 0 - -
OR	R	#N	3	10	dst<-dst OR src	- ^ ^ 0 - -
OR	(rr)	#N	3	16	dst<-dst OR src	- ^ ^ 0 - -
OR	NN	#N	5	24	dst<-dst OR src	- ^ ^ 0 - -
OR	(rr)	(rr)	3	20	dst<-dst OR src	- ^ ^ 0 - -
OR	(RR)	(rr)	3	20	dst<-dst OR src	- ^ ^ 0 - -

Note 1. Refer to ST9 programming manual for detailed information.

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
ORW : Logical OR between two words						
ORW	rr	rr	2	10	dst<-dst OR src	- ^ ^ 0 - -
ORW	RR	RR	3	12	dst<-dst OR src	- ^ ^ 0 - -
ORW	rr	RR	3	12	dst<-dst OR src	- ^ ^ 0 - -
ORW	RR	rr	3	12	dst<-dst OR src	- ^ ^ 0 - -
ORW	rr	(r)	3	14	dst<-dst OR src	- ^ ^ 0 - -
ORW	RR	(r)	3	14	dst<-dst OR src	- ^ ^ 0 - -
ORW	rr	(rr)	2	16	dst<-dst OR src	- ^ ^ 0 - -
ORW	RR	(rr)	3	18	dst<-dst OR src	- ^ ^ 0 - -
ORW	rr	NN	4	22	dst<-dst OR src	- ^ ^ 0 - -
ORW	rr	N(rrx)	4	28	dst<-dst OR src	- ^ ^ 0 - -
ORW	RR	N(rrx)	4	28	dst<-dst OR src	- ^ ^ 0 - -
ORW	rr	NN(rrx)	5	30	dst<-dst OR src	- ^ ^ 0 - -
ORW	RR	NN(rrx)	5	30	dst<-dst OR src	- ^ ^ 0 - -
ORW	rr	rr(rrx)	3	26	dst<-dst OR src	- ^ ^ 0 - -
ORW	rr	(rr)+	3	22	dst<-dst OR src rr<-rr+2	- ^ ^ 0 - -
ORW	RR	(rr)+	3	22	dst<-dst OR src rr<-rr+2	- ^ ^ 0 - -
ORW	rr	-(rr)	3	24	rr<-rr-2 dst<-dst OR src	- ^ ^ 0 - -
ORW	RR	-(rr)	3	24	rr<-rr-2 dst<-dst OR src	- ^ ^ 0 - -
ORW	(r)	rr	3	14	dst<-dst OR src	- ^ ^ 0 - -
ORW	(r)	RR	3	14	dst<-dst OR src	- ^ ^ 0 - -
ORW	(rr)	rr	2	30	dst<-dst OR src	- ^ ^ 0 - -
ORW	(rr)	RR	3	30	dst<-dst OR src	- ^ ^ 0 - -
ORW	(rr)+	rr	3	32	dst<-dst OR src rr<-rr+2	- ^ ^ 0 - -
ORW	(rr)+	RR	3	32	dst<-dst OR src rr<-rr+2	- ^ ^ 0 - -
ORW	NN	rr	4	32	dst<-dst OR src	- ^ ^ 0 - -
ORW	N(rrx)	rr	4	38	dst<-dst OR src	- ^ ^ 0 - -
ORW	N(rrx)	RR	4	38	dst<-dst OR src	- ^ ^ 0 - -
ORW	NN(rrx)	rr	5	38	dst<-dst OR src	- ^ ^ 0 - -
ORW	NN(rrx)	RR	5	38	dst<-dst OR src	- ^ ^ 0 - -
ORW	rr(rrx)	rr	3	34	dst<-dst OR src	- ^ ^ 0 - -
ORW	-(rr)	rr	3	32	rr<-rr-2 dst<-dst OR src	- ^ ^ 0 - -
ORW	-(rr)	RR	3	32	rr<-rr-2 dst<-dst OR src	- ^ ^ 0 - -
ORW	rr	#NN	4	14	dst<-dst OR src	- ^ ^ 0 - -
ORW	RR	#NN	4	14	dst<-dst OR src	- ^ ^ 0 - -
ORW	(rr)	#NN	4	32	dst<-dst OR src	- ^ ^ 0 - -
ORW	NN	#NN	6	36	dst<-dst OR src	- ^ ^ 0 - -
ORW	N(rrx)	#NN	5	36	dst<-dst OR src	- ^ ^ 0 - -
ORW	NN(rrx)	#NN	6	38	dst<-dst OR src	- ^ ^ 0 - -
ORW	(rr)	(rr)	2	32	dst<-dst OR src	- ^ ^ 0 - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
PEA : Push effective address on system stack						
PEA		N(rrx)	4	20	SSP<-USP-2, (SSP)<-rrx+N	- - - - - -
PEA		NN(rrx)	5	26	SSP<-USP-2, (SSP)<-rrx+N	- - - - - -
PEA		N(RRx)	4	20	SSP<-USP-2, (SSP)<-RRx+N	- - - - - -
PEA		NN(RRx)	5	26	SSP<-USP-2, (SSP)<-RRx+N	- - - - - -
PEAU : Push effective address on user stack						
PEAU		N(rrx)	4	20	USP<-USP-2, (USP)<-rrx+N	- - - - - -
PEAU		NN(rrx)	5	26	USP<-USP-2, (USP)<-rrx+N	- - - - - -
PEAU		N(RRx)	4	20	USP<-USP-2, (USP)<-RRx+N	- - - - - -
PEAU		NN(RRx)	5	26	USP<-USP-2, (USP)<-RRx+N	- - - - - -
POP : Pop system stack						
POP	r		2	10	dst<-(SSP), SSP<-SSP+1	- - - - - -
POP	R		2	10	dst<-(SSP), SSP<-SSP+1	- - - - - -
POP	(r)		2	10	dst<-(SSP), SSP<-SSP+1	- - - - - -
POP	(R)		2	10	dst<-(SSP), SSP<-SSP+1	- - - - - -
POPU : Pop user stack						
POPU	r		2	10	dst<-(USP), USP<-USP+1	- - - - - -
POPU	R		2	10	dst<-(USP), USP<-USP+1	- - - - - -
POPU	(r)		2	10	dst<-(USP), USP<-USP+1	- - - - - -
POPU	(R)		2	10	dst<-(USP), USP<-USP+1	- - - - - -
POPUW : Pop word from user stack						
POPUW	rr		2	14	dst<-(USP), USP<-USP+2	- - - - - -
POPUW	RR		2	14	dst<-(USP), USP<-USP+2	- - - - - -
POPW : Pop word from system stack						
POPW	rr		2	14	dst<-(SSP), SSP<-SSP+2	- - - - - -
POPW	RR		2	14	dst<-(SSP), SSP<-SSP+2	- - - - - -
PUSH : Push system stack						
PUSH		r	2	10	SSP<-SSP-1, (SSP)<-src	- - - - - -
PUSH		R	2	10	SSP<-SSP-1, (SSP)<-src	- - - - - -
PUSH		(r)	2	10	SSP<-SSP-1, (SSP)<-src	- - - - - -
PUSH		(R)	2	10	SSP<-SSP-1, (SSP)<-src	- - - - - -
PUSH		#N	3	16	SSP<-SSP-1, (SSP)<-src	- - - - - -
PUSHU : Push user stack						
PUSHU		r	2	10	USP<-USP-1, (USP)<-src	- - - - - -
PUSHU		R	2	10	USP<-USP-1, (USP)<-src	- - - - - -
PUSHU		(r)	2	10	USP<-USP-1, (USP)<-src	- - - - - -
PUSHU		(R)	2	10	USP<-USP-1, (USP)<-src	- - - - - -
PUSHU		#N	3	16	USP<-USP-1, (USP)<-src	- - - - - -
PUSHUW : Push word on user stack						
PUSHUW		rr	2	12	USP<-USP-2, (USP)<-src	- - - - - -
PUSHUW		RR	2	12	USP<-USP-2, (USP)<-src	- - - - - -
PUSHUW		#NN	4	20	USP<-USP-2, (USP)<-src	- - - - - -
PUSHW : Push Word on System Stack						
PUSHW		rr	2	12	SSP<-SSP-2, (SSP)<-src	- - - - - -
PUSHW		RR	2	12	SSP<-SSP-2, (SSP)<-src	- - - - - -
PUSHW		#NN	4	20	SSP<-SSP-2, (SSP)<-src	- - - - - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
RCF : Reset carry flag						
RCF			1	6	C <- 0	0 - - - - -
RET : Return from subroutine						
RET			1	12	PC <- (SSP), SSP <- SPP+2	- - - - - -
RLC : Rotate left through carry						
RLC	r		2	6	dst(0)<-C, C<-dst(7) dst(n+1)<-dst(n) n=0-6	^ ^ ^ ^ - -
RLC	R		2	6	" "	^ ^ ^ ^ - -
RLC	(r)		2	6	" "	^ ^ ^ ^ - -
RLC	(R)		2	6	" "	^ ^ ^ ^ - -
RLCW : Rotate word left through carry						
RLCW	rr		2	8	dst(0)<-C, C<-dst(15) dst(n+1)<-dst(n) n=0-14	
RLCW	RR		2	8	" "	
ROL : Rotate left						
ROL	r		2	6	C<-dst(7), dst(0)<-dst(7) dst(n+1)<-dst(n) n=0-6	^ ^ ^ ^ - -
ROL	R		2	6	" "	^ ^ ^ ^ - -
ROL	(r)		2	6	" "	^ ^ ^ ^ - -
ROL	(R)		2	6	" "	^ ^ ^ ^ - -
ROR : Rotate right						
ROR	r		2	6	C<-dst(0), dst(7)<-dst(0) dst(n)<-dst(n+1) n=0-6	^ ^ ^ ^ - -
ROR	R		2	6	" "	^ ^ ^ ^ - -
ROR	(r)		2	6	" "	^ ^ ^ ^ - -
ROR	(R)		2	6	" "	^ ^ ^ ^ - -
RRC : Rotate right through carry						
RRC	r		2	6	dst(7)<-C, C<-dst(0) dst(n)<-dst(n+1) n=0-6	^ ^ ^ ^ - -
RRC	R		2	6	" "	^ ^ ^ ^ - -
RRC	(r)		2	6	" "	^ ^ ^ ^ - -
RRC	(R)		2	6	" "	^ ^ ^ ^ - -
RRCW : Rotate word right through carry						
RRCW	rr		2	8	dst(15)<-C, C<-dst(0) dst(n)<-dst(n+1) n=0-14	^ ^ ^ ^ - -
RRCW	RR		2	8	" "	^ ^ ^ ^ - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
SBC : Subtraction of 2 bytes with carry						
SBC	r	r	2	6	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	R	R	3	10	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	r	R	3	10	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	R	r	3	10	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	r	(r)	2	6	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	R	(r)	3	10	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	r	(rr)	3	12	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	R	(rr)	3	12	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	r	NN	4	18	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	r	N(rrx)	4	24	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	R	N(rrx)	4	24	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	r	NN(rrx)	5	26	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	R	NN(rrx)	5	26	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	r	rr(rrx)	3	22	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	r	(rr)+	3	16	dst<-dst-src-C rr<-rr+1	^ ^ ^ ^ 1 ^
SBC	R	(rr)+	3	16	dst<-dst-src-C rr<-rr+1	^ ^ ^ ^ 1 ^
SBC	r	-(rr)	3	16	rr<-rr-1 dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	R	-(rr)	3	16	rr<-rr-1 dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	(r)	r	3	10	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	(r)	R	3	10	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	(rr)	r	3	18	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	(rr)	R	3	18	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	(rr)+	r	3	22	dst<-dst-src-C rr<-rr+1	^ ^ ^ ^ 1 ^
SBC	(rr)+	R	3	22	dst<-dst-src-C rr<-rr+1	^ ^ ^ ^ 1 ^
SBC	NN	r	4	20	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	N(rrx)	r	4	26	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	N(rrx)	R	4	26	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	NN(rrx)	r	5	28	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	NN(rrx)	R	5	28	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	rr(rrx)	r	3	24	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	-(rr)	r	3	22	rr<-rr-1 dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	-(rr)	R	3	22	rr<-rr-1 dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	r	#N	3	10	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	R	#N	3	10	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	(rr)	#N	3	16	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	NN	#N	5	24	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	(rr)	(rr)	3	20	dst<-dst-src-C	^ ^ ^ ^ 1 ^
SBC	(RR)	(rr)	3	20	dst<-dst-src-C	^ ^ ^ ^ 1 ^

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
SBCW : Subtract word with carry						
SBCW	rr	rr	2	10	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	RR	RR	3	12	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	rr	RR	3	12	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	RR	rr	3	12	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	rr	(r)	3	14	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	RR	(r)	3	14	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	rr	(rr)	2	16	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	RR	(rr)	3	18	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	rr	NN	4	22	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	rr	N(rrx)	4	28	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	RR	N(rrx)	4	28	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	rr	NN(rrx)	5	30	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	RR	NN(rrx)	5	30	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	rr	rr(rrx)	3	26	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	rr	(rr)+	3	22	dst<-dst-src-C rr<-rr+2	^ ^ ^ ^ ? ?
SBCW	RR	(rr)+	3	22	dst<-dst+src+C rr<-rr+2	^ ^ ^ ^ ? ?
SBCW	rr	-(rr)	3	24	rr<-rr-2 dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	RR	-(rr)	3	24	rr<-rr-2 dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	(r)	rr	3	14	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	(r)	RR	3	14	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	(rr)	rr	2	30	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	(rr)	RR	3	30	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	(rr)+	rr	3	32	dst<-dst-src-C rr<-rr+2	^ ^ ^ ^ ? ?
SBCW	(rr)+	RR	3	32	dst<-dst-src-C rr<-rr+2	^ ^ ^ ^ ? ?
SBCW	NN	rr	4	32	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	N(rrx)	rr	4	38	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	N(rrx)	RR	4	38	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	NN(rrx)	rr	5	38	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	NN(rrx)	RR	5	38	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	rr(rrx)	rr	3	34	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	-(rr)	rr	3	32	rr<-rr-2 dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	-(rr)	RR	3	32	rr<-rr-2 dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	rr	#NN	4	14	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	RR	#NN	4	14	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	(rr)	#NN	4	32	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	NN	#NN	6	36	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	N(rrx)	#NN	5	36	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	NN(rrx)	#NN	6	38	dst<-dst-src-C	^ ^ ^ ^ ? ?
SBCW	(rr)	(rr)	2	32	dst<-dst-src-C	^ ^ ^ ^ ? ?

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
SCF : Set carry flag						
SCF			1	6	C <- 1	1 - - - - -
SDM : Set data memory						
SDM			1	6	Set Data Memory DP<-1 Note 1	- - - - - -
SLA : Shift left arithmetic						
SLA	r		2	6	dst C<-dst(7), dst (0)<-0 dst(n+1)<-dst(n)n=0-6	^ ^ ^ ^ 0 -
	R		3	10	" "	^ ^ ^ ^ 0 -
	(rr)		3	20	" "	^ ^ ^ ^ 0 -
SLAW : Shift word left arithmetic						
SLAW	rr		2	10	C<-dst(15), dst (0)<-0 dst(n+1)<-dst(n)n=1-14	^ ^ ^ ^ - -
	RR		3	12	" "	^ ^ ^ ^ - -
	(rr)		2	32	" "	^ ^ ^ ^ - -
SPM : Set program memory						
SPM			1	6	Set Program Memory DP<-0 Note 2	- - - - - -
SPP : Set page pointer						
SPP		#N	2	6	Set Page Pointer	- - - - - -
SRA : Shift right arithmetic						
SRA	r		2	6	dst(7)<-dst(7), C<-dst(0) dst(n)<-dst(n+1)n=0-6	^ ^ ^ ^ 0 ^
SRA	R		2	6	" "	^ ^ ^ ^ 0 ^
SRA	(r)		2	6	" "	^ ^ ^ ^ 0 ^
SRA	(R)		2	6	" "	^ ^ ^ ^ 0 ^
SRAW : Shift word right arithmetic						
SRAW	rr		2	6	dst(15)<-dst(15), C<-dst(0) dst(n)<-dst(n+1)n=0-14	^ ^ ^ 0 - -
SRAW	RR		2	8	" "	^ ^ ^ 0 - -

Notes:
1 Following this instruction, all addressing modes referring to address spaces will refer to the Data Space.
2 Following this instruction, all addressing modes referring to address spaces will refer to the Program Space, except for the following instructions which operate with Dataspace independently of the setting of the DP flag ;
PUSH(W)/PUSHU(W), POP(W)/POPU(W), PEA/PEAU, and CALL, RET, IRET and interrupt execution
(assuming the Stack Pointers are not pointing to the Register File).

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
SRP : Set register pointer						
SRP		#N	2	6	Set Register Pointer	- - - - - -
SRP0 : Set register pointer 0						
SRP0		#N	2	6	Set Register Pointer 0	- - - - - -
SRP1 : Set register pointer 1						
SRP1		#N	2	6	Set Register Pointer 1	- - - - - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
SUB : Subtraction of 2 bytes without carry						
SUB	r	r	2	6	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	R	R	3	10	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	r	R	3	10	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	R	r	3	10	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	r	(r)	2	6	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	R	(r)	3	10	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	r	(rr)	3	12	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	R	(rr)	3	12	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	r	NN	4	18	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	r	N(rrx)	4	24	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	R	N(rrx)	4	24	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	r	NN(rrx)	5	26	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	R	NN(rrx)	5	26	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	r	rr(rrx)	3	22	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	r	(rr)+	3	16	dst<-dst-src rr<-rr+1	^ ^ ^ ^ 1 ^
SUB	R	(rr)+	3	16	dst<-dst-src rr<-rr+1	^ ^ ^ ^ 1 ^
SUB	r	-(rr)	3	16	rr<-rr-1 dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	R	-(rr)	3	16	rr<-rr-1 dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	(r)	r	3	10	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	(r)	R	3	10	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	(rr)	r	3	18	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	(rr)	R	3	18	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	(rr)+	r	3	22	dst<-dst-src rr<-rr+1	^ ^ ^ ^ 1 ^
SUB	(rr)+	R	3	22	dst<-dst-src rr<-rr+1	^ ^ ^ ^ 1 ^
SUB	NN	r	4	20	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	N(rrx)	r	4	26	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	N(rrx)	R	4	26	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	NN(rrx)	r	5	28	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	NN(rrx)	R	5	28	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	rr(rrx)	r	3	24	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	-(rr)	r	3	22	rr<-rr-1 dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	-(rr)	R	3	22	rr<-rr-1 dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	r	#N	3	10	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	R	#N	3	10	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	(rr)	#N	3	16	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	NN	#N	5	24	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	(rr)	(rr)	3	20	dst<-dst-src	^ ^ ^ ^ 1 ^
SUB	(RR)	(rr)	3	20	dst<-dst-src	^ ^ ^ ^ 1 ^

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
SUBW : Subtract words						
SUBW	rr	rr	2	10	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	RR	RR	3	12	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	rr	RR	3	12	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	RR	rr	3	12	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	rr	(r)	3	14	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	RR	(r)	3	14	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	rr	(rr)	2	16	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	RR	(rr)	3	18	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	rr	NN	4	22	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	rr	N(rrx)	4	28	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	RR	N(rrx)	4	28	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	rr	NN(rrx)	5	30	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	RR	NN(rrx)	5	30	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	rr	rr(rrx)	3	26	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	rr	(rr)+	3	22	dst<-dst-src rr<-rr+2	^ ^ ^ ^ ? ?
SUBW	RR	(rr)+	3	22	dst<-dst-src rr<-rr+2	^ ^ ^ ^ ? ?
SUBW	rr	-(rr)	3	24	rr<-rr-2 dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	RR	-(rr)	3	24	rr<-rr-2 dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	(r)	rr	3	14	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	(r)	RR	3	14	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	(rr)	rr	2	30	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	(rr)	RR	3	30	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	(rr)+	rr	3	32	dst<-dst-src rr<-rr+2	^ ^ ^ ^ ? ?
SUBW	(rr)+	RR	3	32	dst<-dst-src rr<-rr+2	^ ^ ^ ^ ? ?
SUBW	NN	rr	4	32	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	N(rrx)	rr	4	38	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	N(rrx)	RR	4	38	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	NN(rrx)	rr	5	38	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	NN(rrx)	RR	5	38	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	rr(rrx)	rr	3	34	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	-(rr)	rr	3	32	rr<-rr-2 dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	-(rr)	RR	3	32	rr<-rr-2 dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	rr	#NN	4	14	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	RR	#NN	4	14	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	(rr)	#NN	4	32	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	NN	#NN	6	36	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	N(rrx)	#NN	5	36	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	NN(rrx)	#NN	6	38	dst<-dst-src	^ ^ ^ ^ ? ?
SUBW	(rr)	(rr)	2	32	dst<-dst-src	^ ^ ^ ^ ? ?
SWAP : Swap nibbles						
SWAP	r		2	8	dst(0-3)<—->dst(4-7)	? ^ ^ ? - -
SWAP	R		2	8	dst(0-3)<—->dst(4-7)	? ^ ^ ? - -
SWAP	(r)		2	8	dst(0-3)<—->dst(4-7)	? ^ ^ ? - -
SWAP	(R)		2	8	dst(0-3)<—->dst(4-7)	? ^ ^ ? - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
TCM : Test and complement byte under mask						
TCM	r	r	2	6	NOT dst AND src	- ^ ^ 0 - -
TCM	R	R	3	10	NOT dst AND src	- ^ ^ 0 - -
TCM	r	R	3	10	NOT dst AND src	- ^ ^ 0 - -
TCM	R	r	3	10	NOT dst AND src	- ^ ^ 0 - -
TCM	r	(r)	2	6	NOT dst AND src	- ^ ^ 0 - -
TCM	R	(r)	3	10	NOT dst AND src	- ^ ^ 0 - -
TCM	r	(rr)	3	12	NOT dst AND src	- ^ ^ 0 - -
TCM	R	(rr)	3	12	NOT dst AND src	- ^ ^ 0 - -
TCM	r	NN	4	18	NOT dst AND src	- ^ ^ 0 - -
TCM	r	N(rrx)	4	24	NOT dst AND src	- ^ ^ 0 - -
TCM	R	N(rrx)	4	24	NOT dst AND src	- ^ ^ 0 - -
TCM	r	NN(rrx)	5	26	NOT dst AND src	- ^ ^ 0 - -
TCM	R	NN(rrx)	5	26	NOT dst AND src	- ^ ^ 0 - -
TCM	r	rr(rrx)	3	22	NOT dst AND src	- ^ ^ 0 - -
TCM	r	(rr)+	3	16	NOT dst AND src rr<-rr+1	- ^ ^ 0 - -
TCM	R	(rr)+	3	16	NOT dst AND src rr<-rr+1	- ^ ^ 0 - -
TCM	r	-(rr)	3	16	rr<-rr-1 NOT dst AND src	- ^ ^ 0 - -
TCM	R	-(rr)	3	16	rr<-rr-1 NOT dst AND src	- ^ ^ 0 - -
TCM	(r)	r	3	10	NOT dst AND src	- ^ ^ 0 - -
TCM	(r)	R	3	10	NOT dst AND src	- ^ ^ 0 - -
TCM	(rr)	r	3	18	NOT dst AND src	- ^ ^ 0 - -
TCM	(rr)	R	3	18	NOT dst AND src	- ^ ^ 0 - -
TCM	(rr)+	r	3	22	NOT dst AND src rr<-rr+1	- ^ ^ 0 - -
TCM	(rr)+	R	3	22	dst<-ds AND src rr<-rr+1	- ^ ^ 0 - -
TCM	NN	r	4	20	NOT dst AND src	- ^ ^ 0 - -
TCM	N(rrx)	r	4	26	NOT dst AND src	- ^ ^ 0 - -
TCM	N(rrx)	R	4	26	NOT dst AND src	- ^ ^ 0 - -
TCM	NN(rrx)	r	5	28	NOT dst AND src	- ^ ^ 0 - -
TCM	NN(rrx)	R	5	28	NOT dst AND src	- ^ ^ 0 - -
TCM	rr(rrx)	r	3	24	NOT dst AND src	- ^ ^ 0 - -
TCM	-(rr)	r	3	22	NOT dst AND src rr<-rr-1	- ^ ^ 0 - -
TCM	-(rr)	R	3	22	NOT dst AND src rr<-rr-1	- ^ ^ 0 - -
TCM	r	#N	3	10	NOT dst AND src	- ^ ^ 0 - -
TCM	R	#N	3	10	NOT dst AND src	- ^ ^ 0 - -
TCM	(rr)	#N	3	16	NOT dst AND src	- ^ ^ 0 - -
TCM	NN	#N	5	22	NOT dst AND src	- ^ ^ 0 - -
TCM	(rr)	(rr)	3	18	NOT dst AND src	- ^ ^ 0 - -
TCM	(RR)	(rr)	3	18	NOT dst AND src	- ^ ^ 0 - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
TCMW : Test and complement word under mask						
TCMW	rr	rr	2	10	NOT dst AND src	- ^ ^ 0 - -
TCMW	RR	RR	3	12	NOT dst AND src	- ^ ^ 0 - -
TCMW	rr	RR	3	12	NOT dst AND src	- ^ ^ 0 - -
TCMW	RR	rr	3	12	NOT dst AND src	- ^ ^ 0 - -
TCMW	rr	(r)	3	14	NOT dst AND src	- ^ ^ 0 - -
TCMW	RR	(r)	3	14	NOT dst AND src	- ^ ^ 0 - -
TCMW	rr	(rr)	2	16	NOT dst AND src	- ^ ^ 0 - -
TCMW	RR	(rr)	3	18	NOT dst AND src	- ^ ^ 0 - -
TCMW	rr	NN	4	22	NOT dst AND src	- ^ ^ 0 - -
TCMW	rr	N(rrx)	4	28	NOT dst AND src	- ^ ^ 0 - -
TCMW	RR	N(rrx)	4	28	NOT dst AND src	- ^ ^ 0 - -
TCMW	rr	NN(rrx)	5	30	NOT dst AND src	- ^ ^ 0 - -
TCMW	RR	NN(rrx)	5	30	NOT dst AND src	- ^ ^ 0 - -
TCMW	rr	rr(rrx)	3	26	NOT dst AND src	- ^ ^ 0 - -
TCMW	rr	(rr)+	3	22	NOT dst AND src rr<-rr+2	- ^ ^ 0 - -
TCMW	RR	(rr)+	3	22	NOT dst AND src rr<-rr+2	- ^ ^ 0 - -
TCMW	rr	-(rr)	3	24	rr<-rr-2 NOT dst AND src	- ^ ^ 0 - -
TCMW	RR	-(rr)	3	24	rr<-rr-2 NOT dst AND src	- ^ ^ 0 - -
TCMW	(r)	rr	3	14	NOT dst AND src	- ^ ^ 0 - -
TCMW	(r)	RR	3	14	NOT dst AND src	- ^ ^ 0 - -
TCMW	(rr)	rr	2	30	NOT dst AND src	- ^ ^ 0 - -
TCMW	(rr)	RR	3	28	NOT dst AND src	- ^ ^ 0 - -
TCMW	(rr)+	rr	3	30	NOT dst AND src rr<-rr+2	- ^ ^ 0 - -
TCMW	(rr)+	RR	3	30	NOT dst AND src rr<-rr+2	- ^ ^ 0 - -
TCMW	NN	rr	4	30	NOT dst AND src	- ^ ^ 0 - -
TCMW	N(rrx)	rr	4	36	NOT dst AND src	- ^ ^ 0 - -
TCMW	N(rrx)	RR	4	36	NOT dst AND src	- ^ ^ 0 - -
TCMW	NN(rrx)	rr	5	36	NOT dst AND src	- ^ ^ 0 - -
TCMW	NN(rrx)	RR	5	36	NOT dst AND src	- ^ ^ 0 - -
TCMW	rr(rrx)	rr	3	32	NOT dst AND src	- ^ ^ 0 - -
TCMW	-(rr)	rr	3	30	rr<-rr-2 NOT dst AND src	- ^ ^ 0 - -
TCMW	-(rr)	RR	3	30	rr<-rr-2 NOT dst AND src	- ^ ^ 0 - -
TCMW	rr	#NN	4	14	NOT dst AND src	- ^ ^ 0 - -
TCMW	RR	#NN	4	14	NOT dst AND src	- ^ ^ 0 - -
TCMW	(rr)	#NN	4	30	NOT dst AND src	- ^ ^ 0 - -
TCMW	NN	#NN	6	34	NOT dst AND src	- ^ ^ 0 - -
TCMW	N(rrx)	#NN	5	34	NOT dst AND src	- ^ ^ 0 - -
TCMW	NN(rrx)	#NN	6	36	NOT dst AND src	- ^ ^ 0 - -
TCMW	(rr)	(rr)	2	32	NOT dst AND src	- ^ ^ 0 - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
TM : Test byte under mask						
TM	r	r	2	6	dst AND src	- ^ ^ 0 - -
TM	R	R	3	10	dst AND src	- ^ ^ 0 - -
TM	r	R	3	10	dst AND src	- ^ ^ 0 - -
TM	R	r	3	10	dst AND src	- ^ ^ 0 - -
TM	r	(r)	2	6	dst AND src	- ^ ^ 0 - -
TM	R	(r)	3	10	dst AND src	- ^ ^ 0 - -
TM	r	(rr)	3	12	dst AND src	- ^ ^ 0 - -
TM	R	(rr)	3	12	dst AND src	- ^ ^ 0 - -
TM	r	NN	4	18	dst AND src	- ^ ^ 0 - -
TM	r	N(rrx)	4	24	dst AND src	- ^ ^ 0 - -
TM	R	N(rrx)	4	24	dst AND src	- ^ ^ 0 - -
TM	r	NN(rrx)	5	26	dst AND src	- ^ ^ 0 - -
TM	R	NN(rrx)	5	26	dst AND src	- ^ ^ 0 - -
TM	r	rr(rrx)	3	22	dst AND src	- ^ ^ 0 - -
TM	r	(rr)+	3	16	dst AND src rr<-rr+1	- ^ ^ 0 - -
TM	R	(rr)+	3	16	dst AND -src rr<-rr+1	- ^ ^ 0 - -
TM	r	-(rr)	3	16	rr<-rr-1 dst AND src	- ^ ^ 0 - -
TM	R	-(rr)	3	16	rr<-rr-1 dst AND src	- ^ ^ 0 - -
TM	(r)	r	3	10	dst AND src	- ^ ^ 0 - -
TM	(r)	R	3	10	dst AND src	- ^ ^ 0 - -
TM	(rr)	r	3	18	dst AND src	- ^ ^ 0 - -
TM	(rr)	R	3	18	dst AND src	- ^ ^ 0 - -
TM	(rr)+	r	3	22	dst AND src rr<-rr+1	- ^ ^ 0 - -
TM	(rr)+	R	3	22	dst AND src rr<-rr+1	- ^ ^ 0 - -
TM	NN	r	4	20	dst AND src	- ^ ^ 0 - -
TM	N(rrx)	r	4	26	dst AND src	- ^ ^ 0 - -
TM	N(rrx)	R	4	26	dst AND src	- ^ ^ 0 - -
TM	NN(rrx)	r	5	28	dst AND src	- ^ ^ 0 - -
TM	NN(rrx)	R	5	28	dst AND src	- ^ ^ 0 - -
TM	rr(rrx)	r	3	24	dst AND src	- ^ ^ 0 - -
TM	-(rr)	r	3	22	rr->rr-1 dst AND src	- ^ ^ 0 - -
TM	-(rr)	R	3	22	rr->rr-1 dst AND src	- ^ ^ 0 - -
TM	r	#N	3	10	dst AND src	- ^ ^ 0 - -
TM	R	#N	3	10	dst AND src	- ^ ^ 0 - -
TM	(rr)	#N	3	16	dst AND src	- ^ ^ 0 - -
TM	NN	#N	5	22	dst AND src	- ^ ^ 0 - -
TM	(rr)	(rr)	3	18	dst AND src	- ^ ^ 0 - -
TM	(RR)	(rr)	3	18	dst AND src	- ^ ^ 0 - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
TMW : Test word under mask						
TMW	rr	rr	2	10	dst AND src	- ^ ^ 0 - -
TMW	RR	RR	3	12	dst AND src	- ^ ^ 0 - -
TMW	rr	RR	3	12	dst AND src	- ^ ^ 0 - -
TMW	RR	rr	3	12	dst AND src	- ^ ^ 0 - -
TMW	rr	(r)	3	14	dst AND src	- ^ ^ 0 - -
TMW	RR	(r)	3	14	dst AND src	- ^ ^ 0 - -
TMW	rr	(rr)	2	16	dst AND src	- ^ ^ 0 - -
TMW	RR	(rr)	3	18	dst AND src	- ^ ^ 0 - -
TMW	rr	NN	4	22	dst AND src	- ^ ^ 0 - -
TMW	rr	N(rrx)	4	28	dst AND src	- ^ ^ 0 - -
TMW	RR	N(rrx)	4	28	dst AND src	- ^ ^ 0 - -
TMW	rr	NN(rrx)	5	30	dst AND src	- ^ ^ 0 - -
TMW	RR	NN(rrx)	5	30	dst AND src	- ^ ^ 0 - -
TMW	rr	rr(rrx)	3	26	dst AND src	- ^ ^ 0 - -
TMW	rr	(rr)+	3	22	dst AND src rr<-rr+2	- ^ ^ 0 - -
TMW	RR	(rr)+	3	22	dst AND src rr<-rr+2	- ^ ^ 0 - -
TMW	rr	-(rr)	3	24	rr<-rr-2 dst AND src	- ^ ^ 0 - -
TMW	RR	-(rr)	3	24	rr<-rr-2 dst AND src	- ^ ^ 0 - -
TMW	(r)	rr	3	14	dst AND src	- ^ ^ 0 - -
TMW	(r)	RR	3	14	dst AND src	- ^ ^ 0 - -
TMW	(rr)	rr	2	28	dst AND src	- ^ ^ 0 - -
TMW	(rr)	RR	3	28	dst AND src	- ^ ^ 0 - -
TMW	(rr)+	rr	3	30	dst AND src rr<-rr+2	- ^ ^ 0 - -
TMW	(rr)+	RR	3	30	dst AND src rr<-rr+2	- ^ ^ 0 - -
TMW	NN	rr	4	30	dst AND src	- ^ ^ 0 - -
TMW	N(rrx)	rr	4	36	dst AND src	- ^ ^ 0 - -
TMW	N(rrx)	RR	4	36	dst AND src	- ^ ^ 0 - -
TMW	NN(rrx)	rr	5	36	dst AND src	- ^ ^ 0 - -
TMW	NN(rrx)	RR	5	36	dst AND src	- ^ ^ 0 - -
TMW	rr(rrx)	rr	3	32	dst AND src	- ^ ^ 0 - -
TMW	-(rr)	rr	3	30	rr<-rr-2 dst AND src	- ^ ^ 0 - -
TMW	-(rr)	RR	3	30	rr<-rr-2 dst AND src	- ^ ^ 0 - -
TMW	rr	#NN	4	14	dst AND src	- ^ ^ 0 - -
TMW	RR	#NN	4	14	dst AND src	- ^ ^ 0 - -
TMW	(rr)	#NN	4	30	dst AND src	- ^ ^ 0 - -
TMW	NN	#NN	6	34	dst AND src	- ^ ^ 0 - -
TMW	N(rrx)	#NN	5	34	dst AND src	- ^ ^ 0 - -
TMW	NN(rrx)	#NN	6	36	dst AND src	- ^ ^ 0 - -
TMW	(rr)	(rr)	2	32	dst AND src	- ^ ^ 0 - -
WFI : Wait for Interrupt						
WFI			2	18	wait for interrupt	- - - - - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
XCH : Exchange Register						
XCH	r	r	3	12	dst <-> src	- - - - - -
XCH	R	R	3	12	dst <-> src	- - - - - -
XCH	r	r	3	12	dst <-> src	- - - - - -
XCH	R	R	3	12	dst <-> src	- - - - - -
XOR : Logical exclusive OR						
XOR	r	r	2	6	dst<-dst XOR src	- ^ ^ 0 - -
XOR	R	R	3	10	dst<-dst XOR src	- ^ ^ 0 - -
XOR	r	R	3	10	dst<-dst XOR src	- ^ ^ 0 - -
XOR	R	r	3	10	dst<-dst XOR src	- ^ ^ 0 - -
XOR	r	(r)	2	6	dst<-dst XOR src	- ^ ^ 0 - -
XOR	R	(r)	3	10	dst<-dst XOR src	- ^ ^ 0 - -
XOR	r	(rr)	3	12	dst<-dst XOR src	- ^ ^ 0 - -
XOR	R	(rr)	3	12	dst<-dst XOR src	- ^ ^ 0 - -
XOR	r	NN	4	18	dst<-dst XOR src	- ^ ^ 0 - -
XOR	r	N(rrx)	4	24	dst<-dst XOR src	- ^ ^ 0 - -
XOR	R	N(rrx)	4	24	dst<-dst XOR src	- ^ ^ 0 - -
XOR	r	NN(rrx)	5	26	dst<-dst XOR src	- ^ ^ 0 - -
XOR	R	NN(rrx)	5	26	dst<-dst XOR src	- ^ ^ 0 - -
XOR	r	rr(rrx)	3	22	dst<-dst XOR src	- ^ ^ 0 - -
XOR	r	(rr)+	3	16	dst<-dst XOR src rr->rr+1	- ^ ^ 0 - -
XOR	R	(rr)+	3	16	dst<-dst XOR src rr->rr+1	- ^ ^ 0 - -
XOR	r	-(rr)	3	16	rr->rr-1 dst<-dst XOR src	- ^ ^ 0 - -
XOR	R	-(rr)	3	16	rr->rr-1 dst<-dst XOR src	- ^ ^ 0 - -
XOR	(r)	r	3	10	dst<-dst XOR src	- ^ ^ 0 - -
XOR	(r)	R	3	10	dst<-dst XOR src	- ^ ^ 0 - -
XOR	(rr)	r	3	18	dst<-dst XOR src	- ^ ^ 0 - -
XOR	(rr)	R	3	18	dst<-dst XOR src	- ^ ^ 0 - -
XOR	(rr)+	r	3	22	dst<-dst XOR src rr->rr+1	- ^ ^ 0 - -
XOR	(rr)+	R	3	22	dst<-dst XOR src rr->rr+1	- ^ ^ 0 - -
XOR	NN	r	4	20	dst<-dst XOR src	- ^ ^ 0 - -
XOR	N(rrx)	r	4	26	dst<-dst XOR src	- ^ ^ 0 - -
XOR	N(rrx)	R	4	26	dst<-dst XOR src	- ^ ^ 0 - -
XOR	NN(rrx)	r	5	28	dst<-dst XOR src	- ^ ^ 0 - -
XOR	NN(rrx)	R	5	28	dst<-dst XOR src	- ^ ^ 0 - -
XOR	rr(rrx)	r	3	24	dst<-dst XOR src	- ^ ^ 0 - -
XOR	-(rr)	r	3	22	rr->rr-1 dst<-dst XOR src	- ^ ^ 0 - -
XOR	-(rr)	R	3	22	rr->rr-1 dst<-dst XOR src	- ^ ^ 0 - -
XOR	r	#N	3	10	dst<-dst XOR src	- ^ ^ 0 - -
XOR	R	#N	3	10	dst<-dst XOR src	- ^ ^ 0 - -
XOR	(rr)	#N	3	16	dst<-dst XOR src	- ^ ^ 0 - -
XOR	NN	#N	5	24	dst<-dst XOR src	- ^ ^ 0 - -
XOR	(rr)	(rr)	3	20	dst<-dst XOR src	- ^ ^ 0 - -
XOR	(RR)	(rr)	3	20	dst<-dst XOR src	- ^ ^ 0 - -

INSTRUCTION SUMMARY (Continued)

Mnemo.	dst	src	Bytes	Clock cycles	Operation	Flags C Z S V D H
XORW : Logical exclusive OR between words						
XORW	rr	rr	2	10	dst<-dst XOR src	- ^ ^ 0 - -
XORW	RR	RR	3	12	dst<-dst XOR src	- ^ ^ 0 - -
XORW	rr	RR	3	12	dst<-dst XOR src	- ^ ^ 0 - -
XORW	RR	rr	3	12	dst<-dst XOR src	- ^ ^ 0 - -
XORW	rr	(r)	3	14	dst<-dst XOR src	- ^ ^ 0 - -
XORW	RR	(r)	3	14	dst<-dst XOR src	- ^ ^ 0 - -
XORW	rr	(rr)	2	16	dst<-dst XOR src	- ^ ^ 0 - -
XORW	RR	(rr)	3	18	dst<-dst XOR src	- ^ ^ 0 - -
XORW	rr	NN	4	22	dst<-dst XOR src	- ^ ^ 0 - -
XORW	rr	N(rrx)	4	28	dst<-dst XOR src	- ^ ^ 0 - -
XORW	RR	N(rrx)	4	28	dst<-dst XOR src	- ^ ^ 0 - -
XORW	rr	NN(rrx)	5	30	dst<-dst XOR src	- ^ ^ 0 - -
XORW	RR	NN(rrx)	5	30	dst<-dst XOR src	- ^ ^ 0 - -
XORW	rr	rr(rrx)	3	26	dst<-dst XOR src	- ^ ^ 0 - -
XORW	rr	(rr)+	3	22	dst<-dst XOR src rr<-rr+2	- ^ ^ 0 - -
XORW	RR	(rr)+	3	22	dst<-dst XOR src rr<-rr+2	- ^ ^ 0 - -
XORW	rr	-(rr)	3	24	rr<-rr-2 dst<-dst XOR src	- ^ ^ 0 - -
XORW	RR	-(rr)	3	24	rr<-rr-2 dst<-dst XOR src	- ^ ^ 0 - -
XORW	(r)	rr	3	14	dst<-dst XOR src	- ^ ^ 0 - -
XORW	(r)	RR	3	14	dst<-dst XOR src	- ^ ^ 0 - -
XORW	(rr)	rr	2	30	dst<-dst XOR src	- ^ ^ 0 - -
XORW	(rr)	RR	3	30	dst<-dst XOR src	- ^ ^ 0 - -
XORW	(rr)+	rr	3	32	dst<-dst XOR src rr<-rr+2	- ^ ^ 0 - -
XORW	(rr)+	RR	3	32	dst<-dst XOR src rr<-rr+2	- ^ ^ 0 - -
XORW	NN	rr	4	32	dst<-dst XOR src	- ^ ^ 0 - -
XORW	N(rrx)	rr	4	38	dst<-dst XOR src	- ^ ^ 0 - -
XORW	N(rrx)	RR	4	38	dst<-dst XOR src	- ^ ^ 0 - -
XORW	NN(rrx)	rr	5	38	dst<-dst XOR src	- ^ ^ 0 - -
XORW	NN(rrx)	RR	5	38	dst<-dst XOR src	- ^ ^ 0 - -
XORW	rr(rrx)	rr	3	34	dst<-dst XOR src	- ^ ^ 0 - -
XORW	-(rr)	rr	3	32	rr<-rr-2 dst<-dst XOR src	- ^ ^ 0 - -
XORW	-(rr)	RR	3	32	rr<-rr-2 dst<-dst XOR src	- ^ ^ 0 - -
XORW	rr	#NN	4	14	dst<-dst XOR src	- ^ ^ 0 - -
XORW	RR	#NN	4	14	dst<-dst XOR src	- ^ ^ 0 - -
XORW	(rr)	#NN	4	32	dst<-dst XOR src	- ^ ^ 0 - -
XORW	NN	#NN	6	36	dst<-dst XOR src	- ^ ^ 0 - -
XORW	N(rrx)	#NN	5	36	dst<-dst XOR src	- ^ ^ 0 - -
XORW	NN(rrx)	#NN	6	38	dst<-dst XOR src	- ^ ^ 0 - -
XORW	(rr)	(rr)	2	32	dst<-dst XOR src	- ^ ^ 0 - -

NOTES

NOTES

SALES OFFICES

EUROPE

DENMARK

2730 HERLEV
Herlev Torv, 4
Tel. (45-44) 94.85.33
Telex: 35411
Telefax: (45-44) 948694

FINLAND

LOHJA SF-08150
Ratakatu, 26
Tel. (358-12) 155.11
Telefax. (358-12) 155.66

FRANCE

94253 GENTILLY Cedex
7 - avenue Gallieni - BP. 93
Tel.: (33-1) 47.40.75.75
Telex: 632570 STMHQ
Telefax: (33-1) 47.40.79.10

67000 STRASBOURG
20, Place des Halles
Tel. (33-88) 75.50.66
Telefax: (33-88) 22.29.32

GERMANY

85630 GRASBRUNN
Bretonischer Ring 4
Postfach 1122
Tel.: (49-89) 460060
Telefax: (49-89) 4605454
Teletex: 897107=STDISTR

60327 FRANKFURT
Gutleutstrasse 322
Tel. (49-69) 237492-3
Telefax: (49-69) 231957
Teletex: 6997689=STVBF

30695 HANNOVER 51
Rotenburger Strasse 28A
Tel. (49-511) 615960-3
Teletex: 5118418 CSFBEH
Telefax: (49-511) 6151243

90491 NÜRNBERG 20
Erlenstegenstrasse, 72
Tel.: (49-911) 59893-0
Telefax: (49-911) 5980701

70499 STUTTGART 31
Mittlerer Pfad 2-4
Tel. (49-711) 13968-0
Telefax: (49-711) 8661427

ITALY

20090 ASSAGO (MI)
V.le Milanofiori - Strada 4 - Palazzo A/4/A
Tel. (39-2) 57546.1 (10 linee)
Telex: 330131 - 330141 SGSAGR
Telefax: (39-2) 8250449

40033 CASALECCHIO DI RENO (BO)
Via R. Fucini, 12
Tel. (39-51) 593029
Telex: 512442
Telefax: (39-51) 591305

00161 ROMA
Via A. Torlonia, 15
Tel. (39-6) 8443341
Telex: 620653 SGSATE I
Telefax: (39-6) 8444474

NETHERLANDS

5652 AR EINDHOVEN
Meerenakkerweg 1
Tel.: (31-40) 550015
Telex: 51186
Telefax: (31-40) 528835

SPAIN

08004 BARCELONA
Calle Gran Via Corts Catalanes, 322
6th Floor, 2th Door
Tel. (34-3) 4251800
Telefax: (34-3) 4253674

28027 MADRID
Calle Albacete, 5
Tel. (34-1) 4051615
Telex: 46033 TCCEE
Telefax: (34-1) 4031134

SWEDEN

S-16421 KISTA
Borgarfjordsgatan, 13 - Box 1094
Tel.: (46-8) 7939220
Telex: 12078 THSWS
Telefax: (46-8) 7504950

SWITZERLAND

1218 GRAND-SACONNEX (GENEVA)
Chemin Francois-Lehmann, 18/A
Tel. (41-22) 7986462
Telex: 415493 STM CH
Telefax: (41-22) 7984869

UNITED KINGDOM and EIRE

MARLOW, BUCKS
Planar House, Parkway
Globe Park
Tel.: (44-628) 890800
Telex: 847458
Telefax: (44-628) 890391

AMERICAS

BRAZIL

05413 SÃO PAULO
R. Henrique Schaumann 286-CJ33
Tel.: (55-11) 883-5455
Telex: (391)11-37988 "UMBR BR"
Telefax : (55-11) 282-2367

CANADA

NEPEAN ONTARIO K2H 9C4
301 Moodie Drive Suite 307
Tel.: (613) 829-9944
Telefax: (613) 829-8998

U.S.A.

NORTH & SOUTH AMERICAN
MARKETING HEADQUARTERS
55 Old Bedford Road
Lincoln, MA 01773
Tel.: (617) 259-0300
Telefax: (617) 259-4421

SALES COVERAGE BY STATE

ALABAMA
Huntsville - Tel.: (205) 533-5995
Fax: (205) 533-9320

ARIZONA
Phoenix - Tel.: (602) 867-6217
Fax: (602) 867-6200

CALIFORNIA
Santa Ana - Tel.: (714) 957-6018
Fax: (714) 957-3281
San Jose - Tel.: (408) 452-8585
Fax: (452) 1549

COLORADO
Boulder - Tel.: (303) 449-9000
Fax: (303) 449-9505

FLORIDA
Boca Raton - Tel.: (407) 997-7233
Fax: (407) 997-7554

GEORGIA
Norcross - Tel.: (404) 242-7444
Fax: (404) 368-9439

ILLINOIS
Schaumburg - Tel.: (708) 517-1890
Fax: (708) 517-1899

INDIANA
Kokomo - Tel.: (317) 455-3500
Fax: (317) 455-3400
Indianapolis - Tel.: (317) 575-5520
Fax: (317) 575-8211

MICHIGAN
Livonia - Tel.: (313) 953-1700
Fax: (313) 462-4071

MINNESOTA
Bloomington - Tel.: (612) 944-0098
Fax: (612) 944-0133

NORTH CAROLINA
Cary - Tel.: (919) 469-1311
Fax: (919) 469-4515

NEW JERSEY
Voorhees - Tel.: (609) 772-6222
Fax: (609) 772-6037

NEW YORK
Poughkeepsie - Tel.: (914) 454-8813
Fax: (914) 454-1320

OREGON
Lake Oswego - Tel.: (503) 635-7650

TENNESSEE
Knoxville - Tel.: (615) 524-6239

TEXAS
Austin - Tel.: (512) 502-3020
Fax: (512) 346-6260
Carrollton - Tel.: (214) 466-8844
Fax: (214) 466-8130
Houston - Tel.: (713) 376-9936
Fax: (713) 376-9948

FOR RF AND MICROWAVE
POWER TRANSISTORS CON-
TACT
THE FOLLOWING REGIONAL
OFFICE IN THE U.S.A.

PENNSYLVANIA
Montgomeryville - Tel.: (215) 361-6400
Fax: (215) 361-1293

ASIA / PACIFIC

AUSTRALIA

NSW 2220 HURTSVILLE
Suite 3, Level 7, Otis House
43 Bridge Street
Tel. (61-2) 5803811
Telefax: (61-2) 5806440

HONG KONG

WANCHAI
22nd Floor - Hopewell centre
183 Queen's Road East
Tel. (852) 8615788
Telex: 60955 ESGIES HX
Telefax: (852) 8656589

INDIA

NEW DELHI 110019
Liaison Office
3rd Floor, F-Block
International Trade Tower
Nehru Place
Tel. (91-11) 644-5928/647-9415
Telex: 031-70193 STMI IN
Telefax: (91-11) 6443054

MALAYSIA

SELANGOR, PETALING JAYA 46200
Unit BM-10
PJ Industrial Park
Jalan Kemajuan 12/18
Tel.: (03) 758 1189
Telefax: (03) 758 1179

PULAU PINANG 10400
4th Floor - Suite 4-03
Bangunan FOP-123D Jalan Anson
Tel. (04) 379735
Telefax (04) 379816

KOREA

SEOUL 121
8th floor Shinwon Building
823-14, Yuksam-Dong
Kang-Nam-Gu
Tel. (82-2) 553-0399
Telex: SGSKOR K29998
Telefax: (82-2) 552-1051

SINGAPORE

SINGAPORE 2056
28 Ang Mo Kio - Industrial Park 2
Tel. (65) 4821411
Telex: RS 55201 ESGIES
Telefax: (65) 4820240

TAIWAN

TAIPEI
11th Floor
105, Section 2 Tun Hua South Road
Tel. (886-2) 755-4111
Telex: 10310 ESGIE TW
Telefax: (886-2) 755-4008

THAILAND

BANGKOK 10110
54 Asoke Road
Sukhumvit 21
Tel.: (662) 260 7870
Telefax: (662) 260 7871

JAPAN

TOKYO 108
Nisseki - Takanawa Bld. 4F
2-18-10 Takanawa
Minato-Ku
Tel. (81-3) 3280-4121
Telefax: (81-3) 3280-4131

Recycled and chlorine free paper