COMPUTER GRAPHICS

APPLICATION MANUAL

1st EDITION

JUNE 1993

INMOS Limited is a member of the SGS-THOMSON Microelectronics Group.

SGS-THOMSON reserves the right to make changes in specifications at any time and without notice. The information furnished by SGS-THOMSON in this publication is believed to be accurate; however, no responsibility is assumed for its use, nor for any infringement of patents or other rights of third parties resulting from its use. No licence is granted under any patents, trademarks or other rights of SGS-THOMSON.

Document number: 72 TRN 250 00

Order Code: AMCOMGRAST/1

TABLE OF CONTENTS

INTRODUCTION

The Computer Graphics Applications Notebook contains Application Notes intended for system designers using the SGS-THOMSON range of computer graphics devices. A brief summary of the Application Notes included in this book is given later in this introduction.

This Applications Notebook should be used in conjunction with the *Computer Graphics Databook*, 3rd edition, SGS-THOMSON, November 1992 (Order code DBGRAPHICST/2).

THE COMPANY

SGS-THOMSON is an international semiconductor company founded in 1987 as a result of the merger of Thomson Semiconducteurs and SGS Microelettronica. In April 1989, SGS-THOMSON further enhanced its international position with the acquisition of INMOS, a British company with leading MOS technology. The part of INMOS responsible for the computer graphics products described in this book is now the Graphics Business Unit of SGS-THOMSON.

SGS-THOMSON has earned a leading position in the world semiconductor market with its rich technological background, considerable production resources, and broad product range which covers all sectors of advanced electronics. According to Dataquest, the most respected market analysts in the semiconductor field, in 1992 SGS-THOMSON, with a sales growth of 12%, outgrew the market growth by two points and with sales over 1.6B US$ confirmed its position of 13th in the worldwide rankings and second European in the worldwide arena.

The combined forces of the Group total 15 production locations, 9 advanced research and development centers, 25 design centers, 44 direct sales offices in 21 countries, and over 600 representatives and distributors.

To guarantee continued technological development and consistently leading edge products, SGS-THOMSON each year invests close to 20% of its sales figures in R&D and is a key player in Europe's advanced technology research programs.

THE PRODUCTS

SGS-THOMSON's computer graphics devices provide, at a basic level, the data conversion and analog output stage to drive a color monitor. The more advanced products integrate further stages and functions of the graphics subsystem. The products cover a wide spectrum of performance and applications in PCs and workstations.

PALETTE-DACS

Palette-DACs are primarily used in PCs which provide VGA standard graphics capability, and in VGA add-in boards. The IMS G171, which rapidly became the industry standard for VGA graphics systems, was adopted by IBM for use in the IBM PS/2 range of PCs. SGS-THOMSON has now shipped well in excess of 30 million Palette-DACs.

The recently introduced IMS G173 and G174 support the high color and true color modes now offered by many VGA controllers. They allow mixed format display onscreen, making them ideal for windowing applications, and increase the number of simultaneously displayable colors from the 256 of standard VGA to 32K (16-bit) or 16 million (24-bit).

SGS-THOMSON also offers a “semi-custom” facility for graphics products using a library of dedicated graphics cells, including PLL, DAC and RAM macro-cells. The customer interface for this can range from complete customer design to SGS-THOMSON supplied turnkey solutions. This technique is used also for new standard products, including the STG 1700, a 16-bit pixel interface Palette-DAC featuring 135MHz video speed (supporting 1280 × 1024 resolution at 76Hz screen refresh) using an on-chip clock doubler (therefore a simple 67.5MHz pixel interface).

COLOR VIDEO CONTROLLERS

The Color Video Controller (CVC) takes functional integration a significant step further, and represents a complete graphics subsystem on a chip. The devices integrate a triple DAC and look up table with a programmable video timing generator, framestore manager, phase locked loop for clock generation, pixel multiplexer and hardware cursor. Applications are mainly in high resolution PC add-in boards and processor direct graphics PCs.

FUTURE GRAPHICS PRODUCTS

SGS-THOMSON is constantly upgrading and developing its product ranges. The Company will continue to develop highly integrated and cost effective products for use in color graphics systems in response to customer requirements and evolving markets, targeting especially the volume PC markets, and striking a balance between new product innovation and established industry standards.

SUMMARY OF APPLICATION NOTES

PALETTE-DACS

1. IMS G173/4 compatibility and hardware design

This note highlights a number of issues relating to the use of the IMS G174. BIOS support, Palette-DAC identification, competitors' part functionality and compatibility and layout issues are discussed.

2. Designing with the IMS G177

This note shows how VGA chipsets control the power-down states of the IMS G177. A brief competitive analysis of alternative devices is also given.

COLOR VIDEO CONTROLLERS

3. 32-bit Color Video Controller upgrade compatibility: a design guide

This Application Note overviews the implementation of the IMS G332 Color Video Controller in graphics designs requiring upgradability to the new IMS G335 controller. New features discussed include split SAM support, asynchronous micro port and the new pixel sampling mechanism allowing 1280×1024 15/16-bit pixels to be displayed at a 76Hz refresh rate. General design implementations include single and multiple VRAM banks, bank interleaving for improving VRAM SAM access latency with and without pixel multiplexors (using Serial Output Enable).

4. Migrating G300x designs to a G335 solution

This note identifies the changes required in converting designs currently using the now obsolete IMS G300 to systems with similar functionality but offering improved or more features using an IMS G335. Micro port, split SAM transfer and software issues are discussed.

5. Hardware design with CVCs

This note discusses and provides recommendations on the analog design, circuit layout and routing of CVC based graphics printed circuit boards (PCBs). These recommendations aim to minimize interactions between digital and analog signals in the same area of a PCB, which in turn provide clean DAC outputs and a screen display free from effects relating to Shift Clock frequencies, DRAM refresh or other digital "events".

6. Compact design considerations when using the IMS G335

This note overviews the design of a microprocessor-based graphics module using the IMS G335 as the local display controller. The design effectively implements a processor direct graphics solution in an area approximately equal to a standard credit card. The note discusses the use of high density/small package EPLDs and other methods of minimizing interface logic. Concentrating on this requirement, the IMS G335's pixel multiplexor is implemented using Serial Output Enables.

This provides a suitable system for sampling 16-bit pixels and displaying frequencies up to 95MHz with the chosen VRAM devices.

7. CVC parameter calculations for driving standard PC displays

This note derives the CVC parameters required for driving standard PC resolutions. Attention is paid to meeting the CVC timing specifications.

8. Driving PAL and NTSC with CVCs

CVCs can drive arbitrary resolutions including both PAL and NTSC. This note derives the parameters required to drive these two standards. Particular attention is paid to meeting the PC display timing specifications.

9. Using the CVC automatic screen refresh

This note explains the programming of the IMS G335 CVC when implementing its automatic screen refresh feature. A variety of screen modes are discussed to clarify some of the issues involved.

10. Using CVCs in slave mode

The requirements of the CVC slave mode are explained, including external (master) HSync and VSync timings, pixel level synchronization, brief genlocking requirements, slave delay, sync pulse level considerations and control register bit settings. Interlaced and non-interlaced operation are covered.

11. Using the on-chip self test capabilities of CVCs

One of the areas of most concern within the computer industry is the area of testability. The ability to accurately test machines accrues benefits to both manufacturer and end user.

To satisfy this need, this Application Note describes how using the on-chip self test capabilities of a CVC, the correct operation of the serial access data path from VRAM right up to the data presented to the triple video DACs within the CVC may be verified. This verification is performed at video data rates.

12. Some common problems driving monitors with CVCs

The connection of multiple sync signals and the lack of HSync pulses during frame flyback (on the IMS G332 and G364 only) can cause problems with certain monitors. This note discusses some solutions.

GENERAL INDEX

Page

PALETTE-DAC APPLICATION NOTES

CVC APPLICATION NOTES

COLOR IMAGE EXAMPLES

COLOR IMAGE EXAMPLES

The color image examples in this section emphasize the features of the SGS-THOMSON IMS G173/4 range of Palette-DACs. Three different resolutions and pixel depths are used: 8 bits per pixel (bpp) at 1024 × 768, 16bpp at 800 × 600, and 24bpp at 640 × 480. A number of factors important to achieving the best possible color graphics display are highlighted, including:

- The effect of different pixel depths when shading
- Using High Color to overcome the limitations of indirect color when using multiple windows
- The effect of different pixel depths and display resolution when rendering
- Mixing different pixel formats using PixMix
- The need for 24 bits per pixel

Figure 1. The IMS G173/4 launch and brochure image

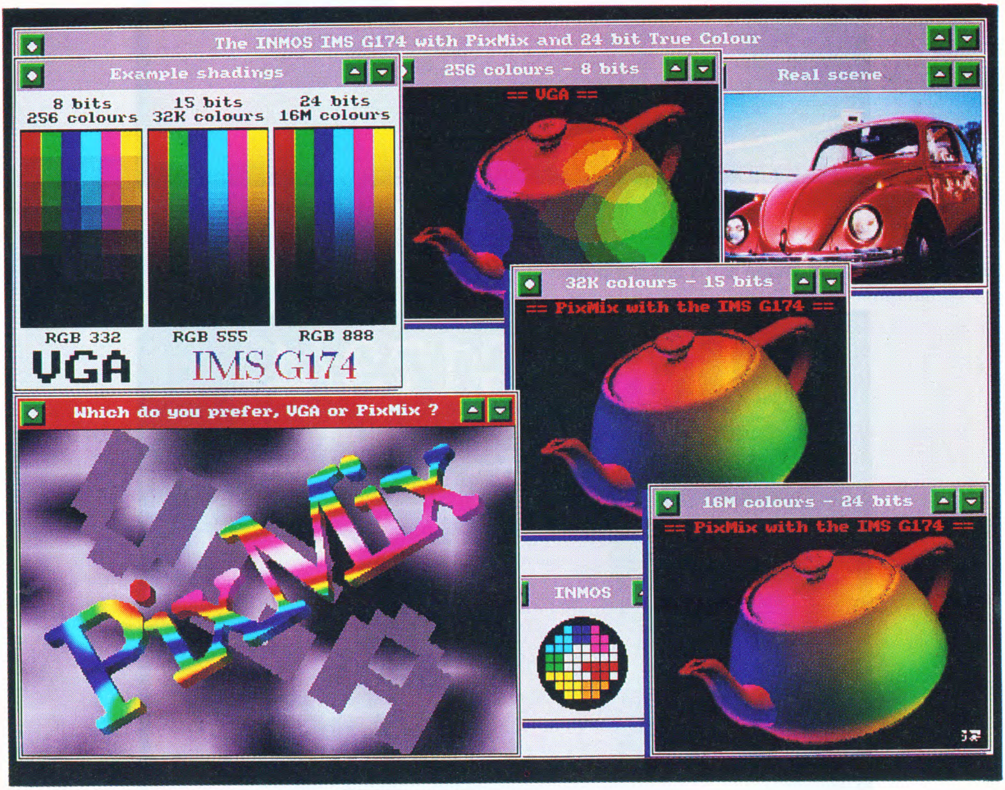


Figure 1 is the 24 bits per pixel image used for the IMS G173/4 brochure and launch. It shows example shadings, a 3D CAD object at 8, 15 and 24 bits per pixel, compares the color "PixMix" with the monochrome "VGA", and includes a real scene at 24 bits per pixel.

Figure 2. Shaded image at 8 bits per pixel

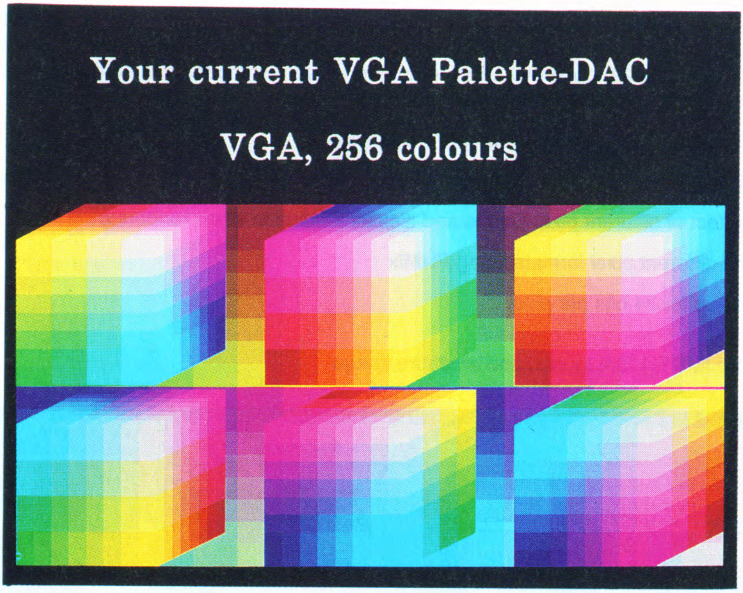


Figure 3. Shaded image at 16 bits per pixel

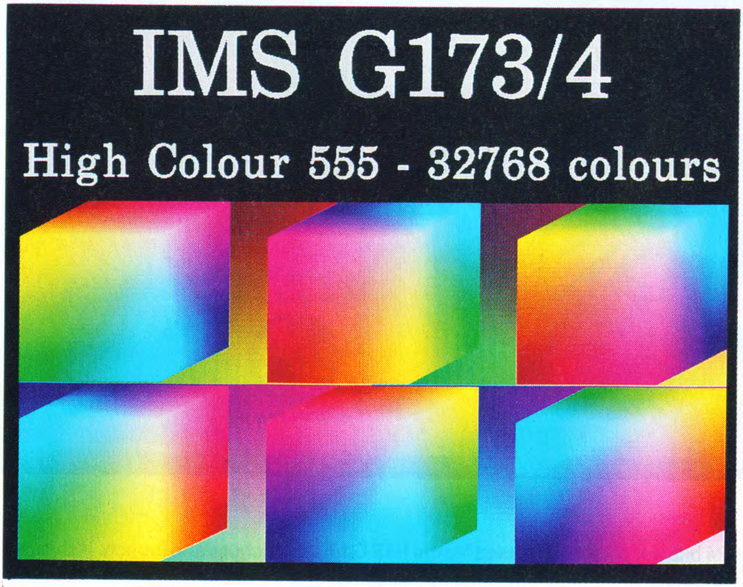
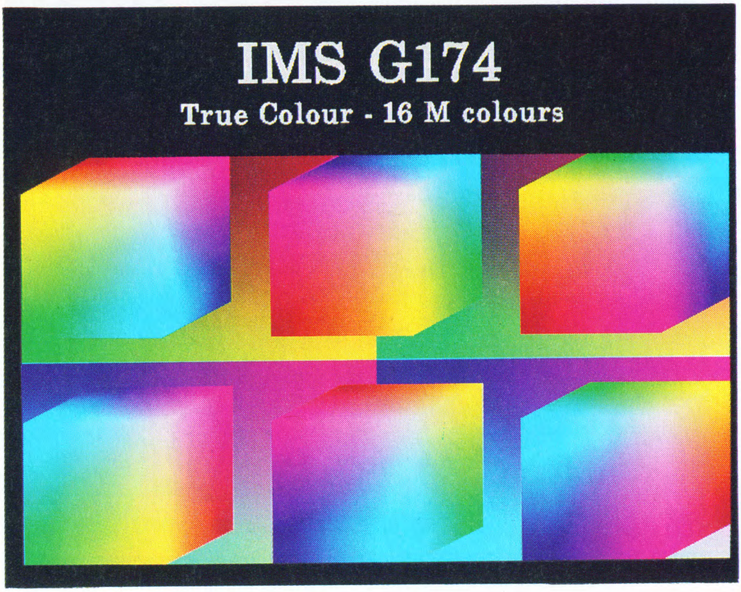


Figure 4. Shaded image at 24 bits per pixel



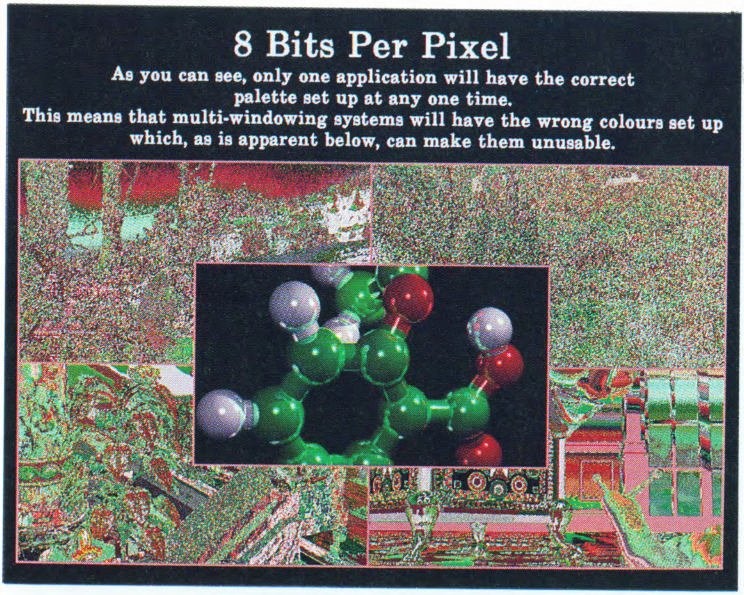
The effect of different pixel depths when shading

Figures 2, 3 and 4 clearly show the differences between 8, 16 and 24 bits per pixel when shading. Note that in Figure 2 using 8 bits per pixel, the shading is quite coarse. Figure 3 shows the improvement achieved using 16 bits per pixel, but 24 bits per pixel, as shown in Figure 4, is really needed for correct shading, .

Examples of using multiple windows

Using 8 bits per pixel can lead to palette problems when using multiple windows. In Figure 5 using 8 bits per pixel, only one window can have its correct palette loaded at any one time. As can be seen, this can make the other windows unusable. Figure 6 shows the advantages of High Color, as all the windows can use their correct colors.

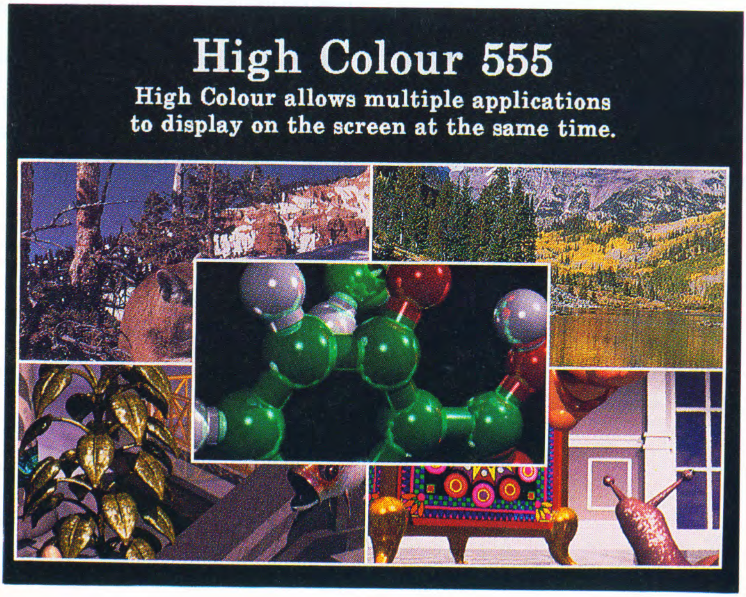
Figure 5. Using multiple windows at 8 bits per pixel



8 Bits Per Pixel

As you can see, only one application will have the correct palette set up at any one time.
This means that multi-windowing systems will have the wrong colours set up which, as is apparent below, can make them unusable.

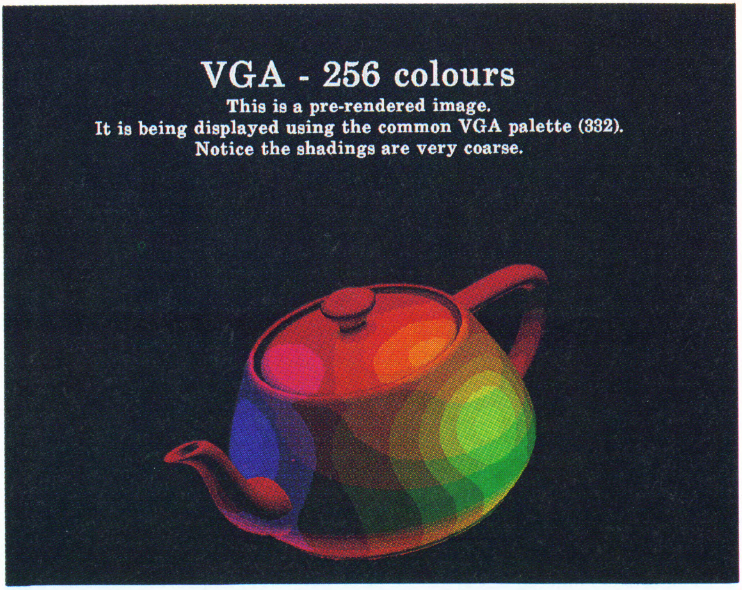
Figure 6. Using multiple windows with High Color



High Colour 555

High Colour allows multiple applications to display on the screen at the same time.

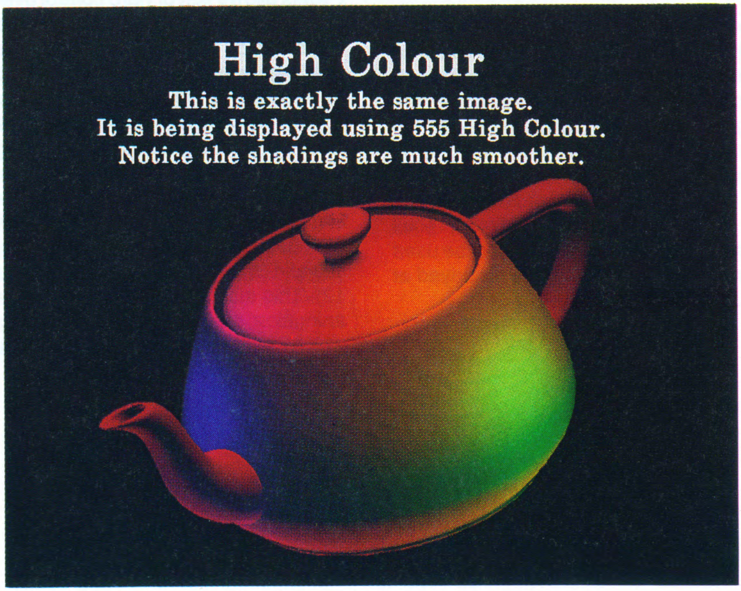
Figure 7. Rendered image at 8 bits per pixel



VGA - 256 colours

This is a pre-rendered image.
It is being displayed using the common VGA palette (332).
Notice the shadings are very coarse.

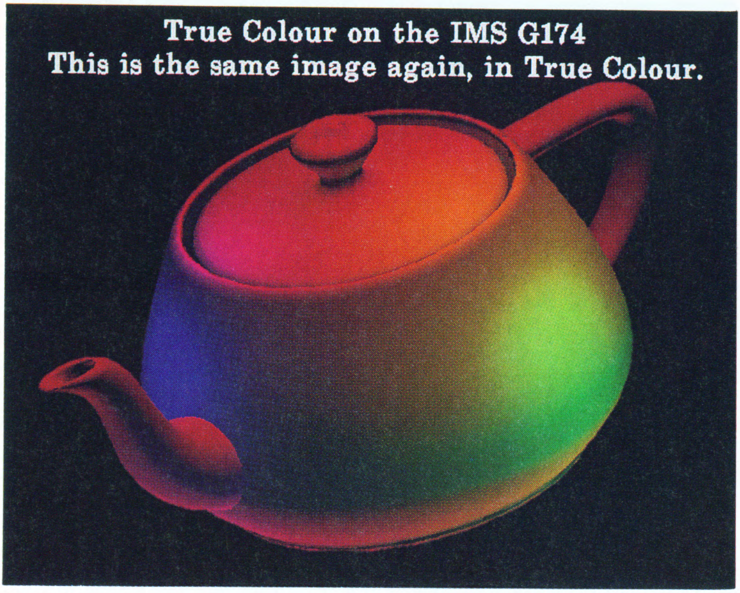
Figure 8. Rendered image at 16 bits per pixel



High Colour

This is exactly the same image.
It is being displayed using 555 High Colour.
Notice the shadings are much smoother.

Figure 9. Rendered image at 24 bits per pixel


True Colour on the IMS G174
This is the same image again, in True Colour.

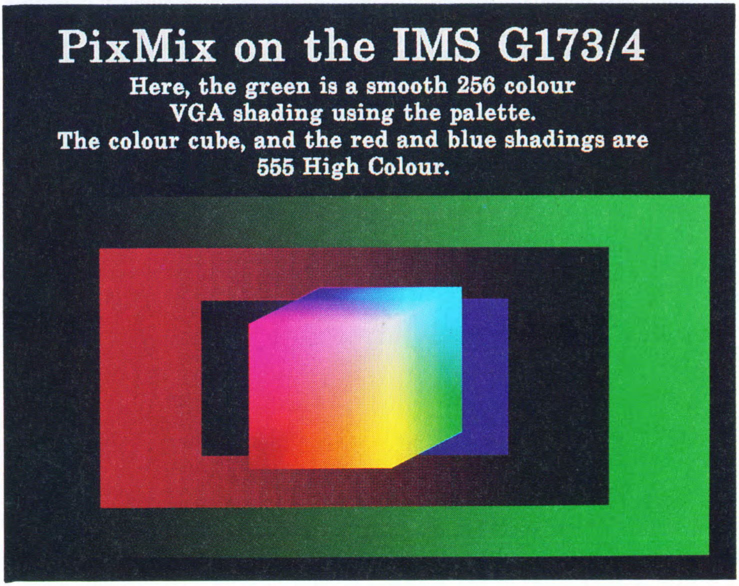
The effect of different pixel depths when rendering

These images show the differences between different pixel depths when rendering the same object (in 3D CAD, for example). Figure 7 shows that the color shading possible with 8 bits per pixel is very limited, resulting in very coarse color graduations. Figure 8 illustrates that while the color shading is dramatically improved by using 16 bits per pixel, 24 bits per pixel are really needed for professional color shading. Note that because different pixel depths are used, different resolutions must be used and consequently the same image is displayed at a different size.

PixMix in operation on the IMS G173/4

PixMix allows soft or hard control over the dynamic mixing of pixel formats. No other Palette-DAC in the world can do this. Figure 10 shows a subtle green shading using the VGA palette at 8 bits per pixel. The red and blue shadings, and the color cube are 5:5:5 High Color. There are advantages of retaining VGA compatibility. The green shading is a 256 color shading – better and smoother than that obtainable using 5:5:5 High Color, which would limit the shading to 5 bits per color field (32 levels of color). Also because the VGA pixels are using the palette, you can change the palette contents (as with Figure 12). This also retains the ability to perform simple animation and special effects using the palette, which cannot be performed using 16-bit High Color or 24-bit True Color.

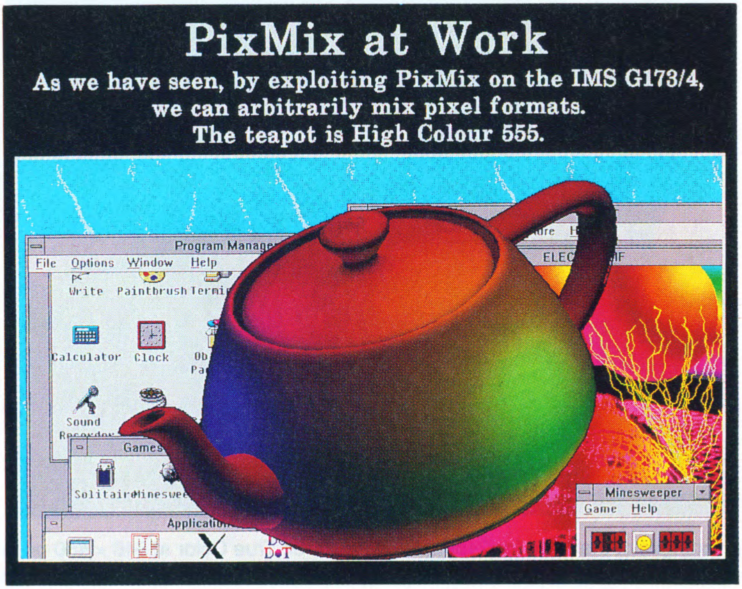
Figure 10. PixMix example showing effect of VGA shading at 8 bits per pixel



PixMix on the IMS G173/4

Here, the green is a smooth 256 colour VGA shading using the palette. The colour cube, and the red and blue shadings are 555 High Colour.

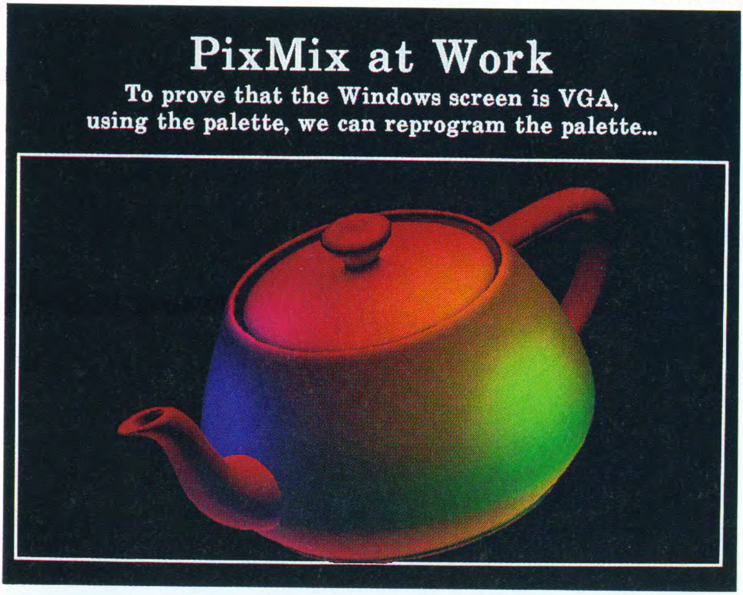
Figure 11. Example of using PixMix to display different pixel formats



PixMix at Work

As we have seen, by exploiting PixMix on the IMS G173/4, we can arbitrarily mix pixel formats. The teapot is High Colour 555.

Figure 12. Example showing the effect of reprogramming the palette when using PixMix



PixMix at Work

To prove that the Windows screen is VGA, using the palette, we can reprogram the palette...

Figure 13. Mount Mandelbrot

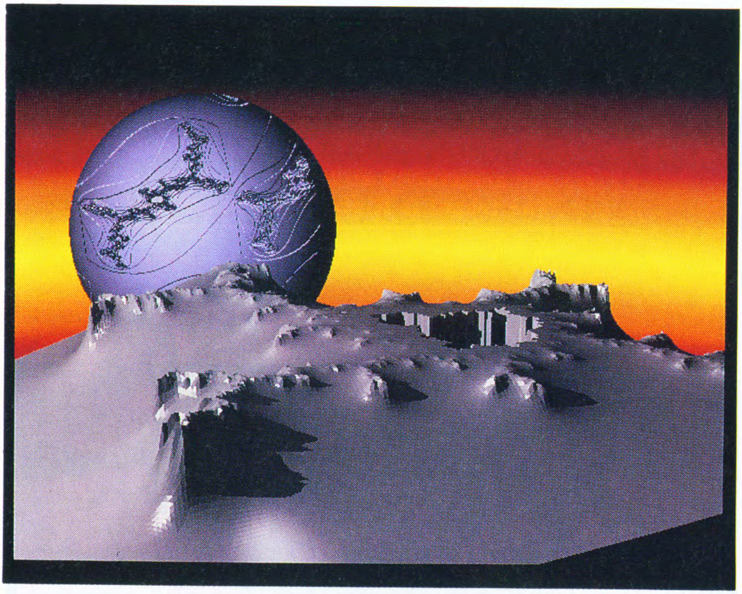


Figure 13 is an example of the need for 24bpp, showing 24-bit True Color at 640 × 480 on the IMS G174. The subtle and realistic shadings of the sky, planet and mountain require the color depth of 24bpp, otherwise color quantization bars would be visible and would detract from the aesthetic quality of such an image.

PALETTE-DAC APPLICATION NOTES

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

IMS G173/4 COMPATIBILITY AND DESIGN CONSIDERATIONS

GRAPHICS APPLICATIONS GROUP, BRISTOL

1.1 INTRODUCTION

This Application Note covers design and compatibility aspects of the SGS-THOMSON IMS G173 and G174 Palette-DACs. The areas covered include functional compatibility with VGA controllers, hardware and software compatibility with Palette-DACs offered by other manufacturers, comparison of IMS G173 and G174 functionality, and design issues including on-chip self test and use of Iref and Vref modes.

Table 1.1.

Part No	Pixel Rates (MHz)	DAC Resolution	Colors	Features	Package
IMS G176	40, 50, 66, 80	6-bits	256	Anti-Sparkle, supports SVGA	28 pin DIP, 32 pin PLCC, 44 pin PLCC
IMS G173	80	6-bits	64K	IMS G176 features plus software PixMix, DAC-gain/fade, 16-bit formats	44 pin PLCC
IMS G174	80,85,100	6/8-bit	16M	IMS G173 features plus hardware PixMix, DAC gain and auto-fade, 24-bit format, XGA	44 pin PLCC

Palette-DAC functionality and features

- **VGA compatibility**

The entire Palette-DAC family from SGS-THOMSON is VGA compatible. In addition, upgrading from the IMS G176 to G173 or G174 ensures backward compatibility in all cases.

- **DAC resolution**

The IMS G174 has an 8-bit DAC resolution. This gives a total color choice of 16 million colors, using 24 bits per pixel. This brings photo-realistic images to the PC screen. All other devices have 6-bit DAC resolution, offering a total choice of 256K colors.

- **High color and true color support**

The IMS G173 and G174 both support 16-bit 5:5:5 (RGB) TARGA, 5:6:5 XGA, 6:6:4 i860 and Select:5:5:5 formats, giving more choice for high color displays. The IMS G174 also supports 8:8:8 in 24-bit true color mode.

- **PixMix**

The PixMix feature, allowing intermixing of high color and pseudo color on a pixel by pixel basis, is described in Section 1.2.5.

- **DAC-fade**

The IMS G173/4 DAC fade feature is described in Section 1.2.6.

1.2 PALETTE-DAC COMPATIBILITY

1.2.1 DAC IDENTIFICATION

Access to extended register space (magic access)

To extend the register space whilst maintaining compatibility with existing Palette-DACs, all high color Palette-DACs use a special access technique (magic access). Reading the Mask Register (address 03C6h) four times in succession causes the next access to be directed to a Pixel Command Register. This Pixel Command Register controls the interpretation of pixel data presented at the pixel port, thus allowing 16 and 24-bit pixels to be constructed. Reading or writing any other valid address resets the special access sequence.

Different makes of high color Palette-DAC differ slightly in their implementation of this special access technique, and it is this aspect, along with other minor differences, which allow high color Palette-DACs to be identified.

In Appendix A an example assembler routine is presented which identifies the following five common makes of high color Palette-DAC:

SGS-THOMSON IMS G174
AT&T 20C491
Sierra SC11483 / SC11486
Acumos ADAC1
Music MU9C4870V

Purpose of Palette-DAC Identification

Most Palette-DACs are compatible at the Pixel Command Register level for 16bpp operation, but there is no standard for 24bpp operation. Each Palette-DAC therefore must potentially be programmed with different values. Identification of the Palette-DAC being used is straightforward and allows a generic BIOS to be used across several products using different Palette-DACs.

Identification of a High Color Palette-DAC

Reading the Mask Register 4 times in succession should cause the next access to be directed to a Pixel Command Register. So the sequence:

```
mov dx, 03c6h
in al, dx            ; 1st mask read
in al, dx            ; 2nd mask read
in al, dx            ; 3rd mask read
in al, dx            ; 4th mask read
in al, dx            ; read pixel command register
```

should read the Pixel Command Register. This register is initialized to 00h on power-up but if the contents of this register cannot be assumed, the sequence:

```
; clean start, mask := ff, cmd reg := 0
mov dx, 03c9h
in al, dx            ; reset seq
mov dx, 03c6h
in al, dx
in al, dx
in al, dx
in al, dx
mov al, 0
out dx, al           ; cmd reg := 0
mov dx, 03c9h
in al, dx            ; reset seq
mov dx, 03c6h
mov al, 0ffh
out dx, al           ; mask := ff
```

will write 0 to the Pixel Command Register (if it exists) and 0FFh to the Mask Register, thus setting the command register and mask to known values. Note the order of this, so that if the part is not a high color Palette-DAC, the mask will still be initialized to 0FFh. Reading the Pixel Command Register after such a sequence should access 0 if the device is a high color Palette-DAC. If 0FFh is read, then the device is a normal Palette-DAC.

The implementation of the special (magic) access sequence necessary to access the Pixel Command Register varies slightly from one make of Palette-DAC to another.

The SGS-THOMSON IMS G173/4 reads the Mask Register four times in order to access the Pixel Command Register, as normal. Following this convention, reading the Pixel Command Register four times in succession allows the next access to be made to the XGA Enable Register. The next access will access the Mask Register again, resetting the magic access sequence. As with normal special (magic) access sequences, accessing any other valid address will reset this sequence.

Sierra Palette-DACs (SC1148X) after four successive reads of the Mask Register allow access to the Pixel Command Register until it is written or another valid address is accessed.

AT&T and ACUMOS Palette-DACs (AT&T 20C491, ACUMOS ADAC1) allow just one access to the Pixel Command Register, before reverting to the Mask Register again.

Music Palette-DACs (MU9C4870V) read the Mask Register three times, then read a device ID byte (088h or 089h), then allow access to the Pixel Command Register. Subsequent reads continue to access the Pixel Command Register until it is written or another valid address is accessed.

Reads of the Mask Register can therefore be summarized (assuming that 0FFh is in the Mask Register, and 0 is in the Command Register, and that the special access sequence has been reset):

Read value sequence	Device type
0FFh 0FFh 0FFh (not 0FFh) 0 0 0 0 0 0 ...	Music MU9C4870V
0FFh 0FFh 0FFh 0FFh 0 0FFh 0FFh ...	ATT 20C491, Acumos ADAC1
0FFh 0FFh 0FFh 0FFh 0 0 0 0 0 0FFh 0FFh	SGS-THOMSON IMS G173/4
0FFh 0FFh 0FFh 0FFh 0 0 0 0 0 0 0 ...	Sierra SC1148X

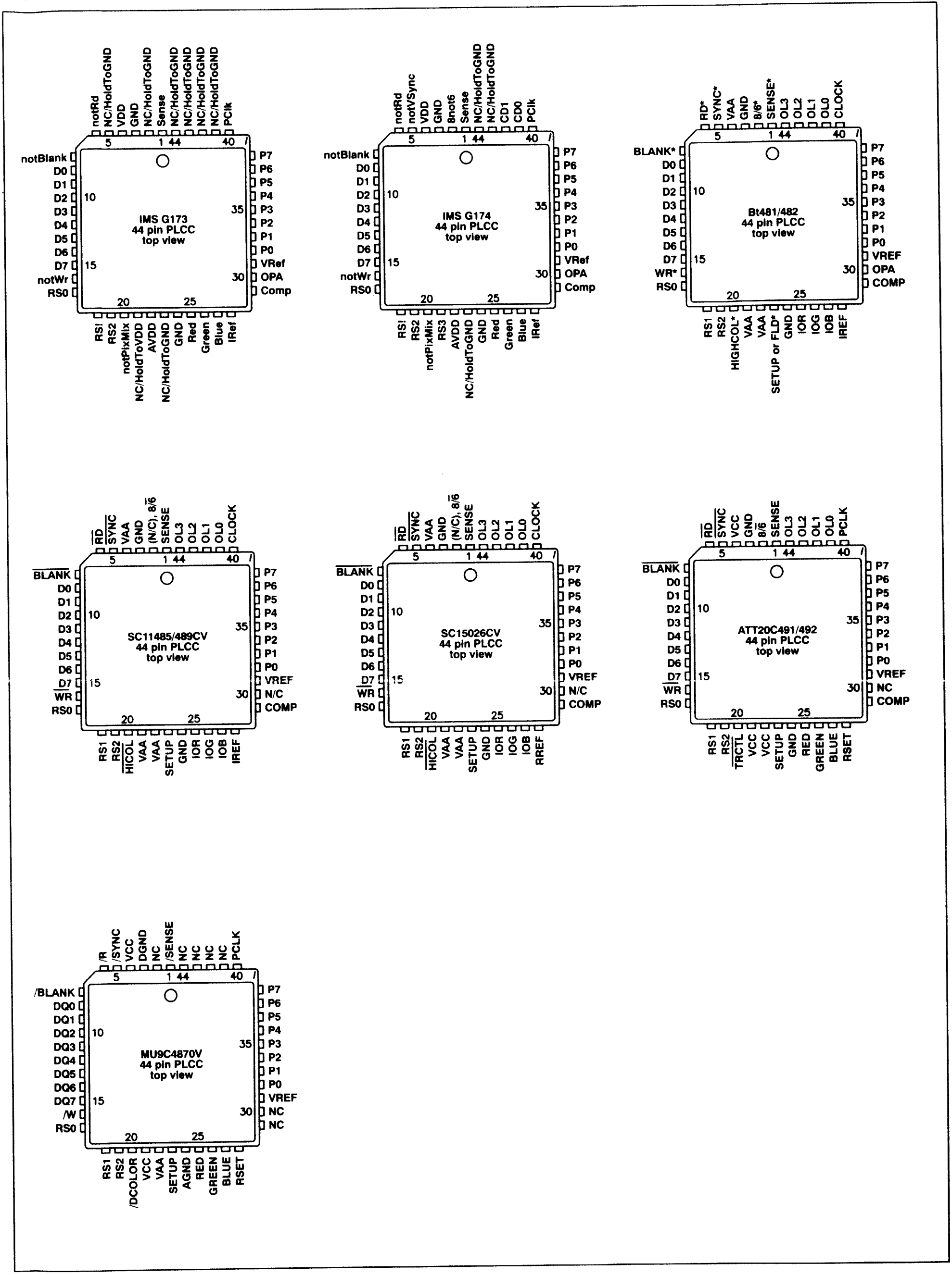
In the example routine shown in Appendix A, these special access sequences are used to identify the Palette-DAC to be one of the five makes listed above. In addition, the AT&T Pixel Command Register bit 1 controls the palette operation mode (whether 6 or 8 bit), and assuming that the **8not6** pin is connected to logic 0, then this feature may be used to identify the AT&T device. Having written 0FFh to the palette, reading in 6-bit mode will read 03Fh, whereas reading in 8-bit mode will read 0FFh.

If 0FFh is written to the ACUMOS Pixel Command Register, then 0Bh is read back.

Summary

Using these characteristics, it is possible to identify which Palette-DAC is being used. This allows a generic BIOS to be used across several products and different Palette-DACs, thus ensuring correct Palette-DAC operation.

Figure 1.1. Palette-DAC pinouts (44 pin PLCC J-bend package)



1.2.2 USING THE IMS G174/3 AS A REPLACEMENT PART FOR OTHER PALETTE-DACS

Replacing the IMS G176 with the IMS G174/3

The IMS G173 and IMS G174 are backwards compatible with the IMS G176. **Iref** (pin 28 on PLCC) and **Comp** (Pin 29) should be connected as shown in Figure 1.4 of Section 1.4 (Analog considerations). This is normally the case in most VGA designs.

Replacing the SC11485/6/7 with the IMS G173

The IMS G173 can be used as a replacement for the SC11485/6/7. All high color modes on the SC11485/6/7 are implemented on the IMS G173.

Overlays and Sync pedestal are not available on the IMS G173.

The Pixel Command Register values are the same for all devices as shown in Table 1.2. Software Palette-DAC identification should identify the IMS G173 as being compatible with the SC11485/6/7.

Replacing the SC11489 with the IMS G174

The SC11489 is backwards compatible with the SC11485/6/7, with the addition of 8-bit DACs. The IMS G174 can be used as a replacement for the SC11489 but does not supply Sync pedestal and overlays. The Pixel Command Register values are identical for both devices as shown in Table 1.2.

The pinout of the SC11489 and the IMS G174 are shown in Figure 1.1.

Replacing the SC15025/SC15026 with the IMS G174

The SC15025/SC15026 devices are very similar to the SC1148X devices, with the addition that they support multiple 32K color palette, selectable by register, and a 24-bit true color facility, which can be gamma corrected. The SC15025/SC15026 has the ability to latch 32-bit data as RGBX, on rising edges of latching clock only and rising and falling edges of latching pixel clock.

The SC15025 is similar to the SC15026 but does not support overlays or sync information on the DAC outputs.

The IMS G174 could be used as a replacement for the SC15025/SC15026 but will not be compatible for the 32-bit RGBX mode, overlays and Sync pedestal.

Replacing the ATT20C492 with the IMS G173

The ATT20C492 is a 6-bit DAC device that supports 6:6:6, 5:6:5 and 5:5:5 high color modes as well as 8-bit pseudo color (256 color). The ATT20C492 has no sync enable on the RGB DAC outputs.

The IMS G173 and ATT20C492 pinouts are shown in Figure 1.1.

The user should be made aware of the differences in values programmed into the Pixel Command registers of the IMS G173 and the ATT20C492 for certain pixel modes shown in Table 1.2.

The IMS G173 does not support 6:6:6 pixel mode.

Replacing the ATT20C491 with the IMS G174

The ATT20C491 is an 8-bit DAC device that supports 24-bit true color, 5:6:5 high color mode and 5:5:5 high color mode. The high color and true color modes can be gamma corrected. Overlays are supported in VGA systems.

The IMS G174 can be used as a replacement part for the ATT20C491 if gamma correction is not required. Again the user should be made aware of the differences of programing the Pixel Command register on the IMS G174 and the ATT20C491 as shown in Table 1.2.

Replacing the Bt481 with the IMS G174

The Bt481 is an 8-bit DAC device that supports 5:6:5 XGA, 5:5:5 high color modes and 8:8:8:8 true color with overlay. It also incorporates a power down mode.

The Bt481 does not provide multiple mask magic access to program internal registers.

The Bt481 does provide cursor support, through the overlay input pins OL3 and OL2. All cursor registers, apart from cursor colors are no longer valid. The Bt481 can be instructed to be put into XGA cursor compatible mode.

The IMS G174 can be used as a replacement for the Bt481 for high color modes. The true color/8-bit overlay mode cannot be implemented on the IMS G174 as dictated by the Bt481. Use of this true color mode with VGA overlay pixel is limited and has a data throughput drawback (one displayable pixel is latched for four clock edges).

Values for the Pixel Command register which set the Bt481 and IMS G174 into high color modes are very similar, with the exception of true color modes which are incompatible between the two devices.

Replacing the Bt482 with the IMS G174

The Bt482 is similar to the Bt481 with the addition of a 32×32×2 onboard cursor which, because of its size, is not XGA compatible.

Most of the implications for using a replacement IMS G174 applying to the Bt481 also apply to the Bt482.

1.2.3 PIXEL MODES

Definition: High Color Mode 1

High Color Mode 1 is defined as pixel sampling of LSB of pixel data on rising edge and MSB of pixel data on falling edge of latching pixel data clock.

Definition: High Color Mode 2

High Color Mode 2 is defined as pixel sampling of LSB and MSB of pixel data on the rising edge of the latching pixel data clock.

Table 1.2. Comparison of pixel modes on IMS G174, Sierra SC 1148X and AT&T 20C491/2

Pixel mode	SGS-THOMSON IMS G174	Sierra SC 1148X	AT&T 20C491/2
8bpp pseudo color	Pixel Command Register = 00h	Command Register = 00h	Control Register = 00h
15bpp high color mode 1	Pixel Command Register = 80h	Command Register = 80h	Control Register = 80h
15bpp high color mode 2	Pixel Command Register = A0h	Command Register = A0h	Control Register = A0h
16bpp high color mode 1	Pixel Command Register = C0h	Command Register = C0h	Not available
16bpp high color mode 2	Pixel Command Register = E0h	Command Register = E0h	Control Register = C0h
6:6:4 mode high color mode 1	Pixel Command Register = D0h	Not available	Not available
6:6:4 mode high color mode 2	Pixel Command Register = F0h	Not available	Not available
15-bit pixel with software PixMix high color mode 1 operation	Pixel Command Register = 88h	Not available	Not available
15-bit pixel with software PixMix high color mode 2 operation	Pixel Command Register = A8h	Not available	Not available
24bpp	Pixel Command register = B0h	Not available	Control Register = E0h

NOTE: To include hardware PixMix function with above pixel modes (except 8bpp) just subtract 80h from values stated above for Pixel Command register

1.2.4 PINOUT DIFFERENCES, MEANINGS AND EXTRA PINS

Differences between Sierra 16-bit Palette-DACs and IMS G174

Setup : Pin 23

This pin tells the device whether to have a 7.5 IRE or 0 IRE blanking pedestal. Equivalent on G174 is **NC**.

OL0–OL3 : Pins 41–44

Overlay pins, a fifteen color overlay input which, when a non-zero value has been latched in, will display an overlay color. These functions are disabled on power up. The pins 42, 41 on the G174 (**CD0-1**) are also ignored on power up and provide cursor plane operation very similar to overlay plane operation.

N/C : Pin 30 (**OPA** on IMS G174)

For Vref usage this is connected to **Comp**, for Iref usage it is unconnected. See Section 1.4.

Vref : Pin 31 For Vref usage

Same as on G174 Vref version, ignored on Iref version

$\overline{\text{HICOL}}$: Pin 20

Equivalent on G174 is **notPixMix**. **notPixMix** is a superset of the $\overline{\text{HICOL}}$ operation in that it can be enabled during active video operation.

Differences between AT&T and IMS G174

OL[3:0] : Pins 41–44, for overlay similar to Sierra operation

TRCTL : Pin 20

Instructs part to select color mode using overlay inputs or control register. Equivalent on IMS G174 is **notPixMix**.

VREF : Pin 31

Vref can be connected by a 0.1µF capacitor to GND, as proposed by AT&T. However **OPA** must be connected to **Comp** as shown in Section 1.4.

1.2.5 PixMix™

What is PixMix?

The IMS G173 and IMS G174 support a facility called PixMix, which allows intermixing of high color and pseudo color on a pixel by pixel basis. This switching can be controlled externally by asserting the **notPixMix** pin on 44 pin PLCC versions of G173 and G174 or through software, by logical detection of bit 15 of a 16-bit high color pixel.

Other similar devices do not provide a PixMix function; many have a pin called HICOL or TRCTL which, when taken active low, place the part in a high color mode, although these devices cannot perform this function on a pixel by pixel basis.

Why should you use PixMix?

In a windowing environment it could be a requirement to have windows which have varying pixel format such as a live video insert that uses 24-bit true color or a 15bpp format as opposed to the current 8 bits per pixel. Using the PixMix functions allows this high color insert to appear on a pixel resolution.

For example hardware PixMix could be controlled with the ET4000/W32 BDE pin number 81 by connecting this pin to the **notPixMix** pin of the G174.

If a hardware PixMix function and a 1 bit wide datastore is used as the PixMix source, then arbitrary shapes could be used to describe a high or true color image. Note, that due to the restriction of pins, the hardware PixMix is not available on the G173 28 pin PDIP.

1.2.6 DAC GAIN AND DAC FADE

When using high color (15/16-bit) or true color (24-bit), pixel information is fed directly to the DACs. Because there is no level of indirection through the palette, fading in or out of the display is not possible by programming the palette. **However the IMS G174 and IMS G173 have a facility that allows the user to fade the display in or out. This can be done manually in steps of 1/15th perceived intensity.**

Alternatively an automatic fading or gain mechanism can be programmed on the IMS G174 which can be set to operate every two or four or six..... or thirty frames. This automatic fading requires the **notVSync** pin to be connected to the relevant signal. Fading in or out of the DACs will stop either when the DACs are fully blanked, normal intensity is reached or the user aborts the cycle by programming the DAC Fade Register (0Ch).

This facility can be useful in slide presentation application programs requiring a subtle fade in and fade out.

1.2.7 XGA REGISTERS

The IMS G174 contains an XGA compatible cursor palette as described in Section 1.2.8. The IMS G174 provides a means of accessing these registers and others through the XGA style of register access (index and data register are provided). Besides the XGA color cursor registers, the IMS G174 also provides the facility for software written for a multi-tasking environment. A means of reading and setting the state of the microport is provided which can be especially useful when software has been interrupted while trying to read or write values to the palette before completing its access.

The XGA index and data registers can either be accessed directly using extra RS lines, if provided by the controller, or through an extended magic access of the Mask Register. If the Mask Register is read eight times after an initialization phase of the microport and the value 04h is written into the Mask register, then the IMS G174 will be put into XGA mode. The Pixel Mask register now becomes the XGA Index register and the Address (read mode) register becomes the XGA Data register. Details of XGA register index addresses are given in the IMS G174 Datasheet.

1.2.8 XGA CURSOR

The cursor facility is only available on the 44 pin PLCC IMS G174. The cursor data inputs **CD0-1** are latched by PClk in the same way as the rest of the Pixel Interface. Some controllers do provide compatible outputs for usage of an XGA compatible cursor e.g. ET4000/W32 provides two pins SP(0:1) which can be directly connected to the IMS G174 **CD0-1**. Three colors are available to display the cursor. Two are defined as registers on the IMS G174 and the third is the complement of the palette or true color.

1.3 USING THE ON-CHIP SELF TEST FACILITIES OF THE IMS G173/4

1.3.1 BENEFITS TO THE MANUFACTURER AND END USER

One of the areas of most concern in the computer industry is the area of testability. The ability to accurately test machines accrues benefits to both the manufacturer and end user.

From the manufacturer's point of view, systems having features embodied within them to facilitate testing, can usually be tested quicker than those which do not. This reduces the manufacturing time with a resultant reduction in build cost and increase in manufacturing throughput.

Systems featuring built in self test features which are accessible by the system operating software, BIOS or device driver can incorporate more thorough system diagnostics and power on self test procedures than would otherwise be the case. The ability to perform such diagnostic procedures increases customer perception about the overall machine quality. It is usually impractical however to incorporate additional logic components on the printed circuit board specifically for test purposes, either because of real estate limitations, cost or the unavailability of test access points (if the required data path is not brought out from a device).

The video subsystem by virtue of its shear complexity and data rate imposes an enormous processing requirement upon the system test function. For graphics systems, the integrity of both the host CPU data path and the pixel port data path must be verified.

1.3.2 IMS G173/4 CHECKSUM REGISTERS

The checksum facility provides a mechanism to validate the data being transferred from the controller to the DACs via the pixel port of the G173/4. This validation is performed at real time pixel data rates.

Three 24-bit checksum registers are provided (one for each color channel) and are located in the XGA register space. They are accessed as described in Section 1.3.4 using the index values shown below in Table 1.3.

Table 1.3. Checksum Registers

Index Value	Bit	R/W	Function
Xi86	7:0	R	RED Byte[0]
Xi87	7:0	R	RED Byte[1]
Xi88	7:0	R	RED Byte[2]
Xi89	7:0	R	GREEN Byte[0]
Xi8A	7:0	R	GREEN Byte[1]
Xi8B	7:0	R	GREEN Byte[2]
Xi8C	7:0	R	BLUE Byte[0]
Xi8D	7:0	R	BLUE Byte[1]
Xi8E	7:0	R	BLUE Byte[2]

The checksum value is dependent upon:

- Usage of hardware and software PixMix
- Cursor position (if used)
- Interlacing (if used)

It is however independent of sync modes and flyback patterns.

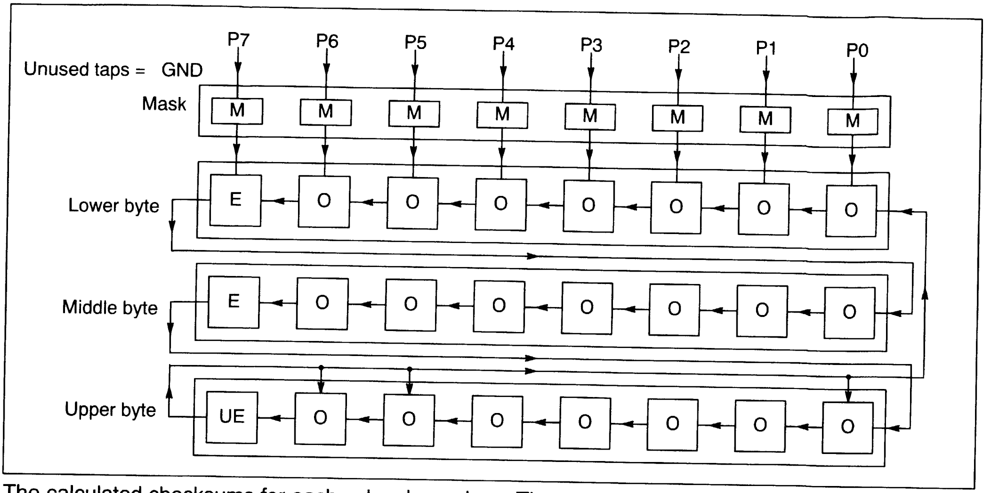
The registers should be read during frame flyback as described in Section 1.3.4. At other times the checksum registers are accumulating and therefore invalid.

Checksum generation

The checksum is generated through a shift register that uses feedback to create a pseudo-random sequence. As each pixel arrives, its 8-bit value is XOR-ed with the lowest 8 bits of the register, and the rest of the register is shifted. The use of feedback minimizes the risk of two errors in the pixel stream cancelling to give a correct checksum value.

The checksum registers are implemented as Linear Feedback Shift Registers (LFSRs) using the Type II design which use internal Exclusive OR feedback taps as shown in Figure 1.2.

Figure 1.2. Implementation of checksum registers



The calculated checksums for each color channel should equal the contents of the appropriate checksum registers in the device. Clearly if PixMix and/or the cursor channel are used then the algorithm used to calculate the expected checksum values will need to be modified accordingly.

Reading and calculating the contents of the three checksum registers is straightforward. Example C/assembler routines are given in Appendix A.3 to explain how certain required tasks are achieved. These routines have deliberately not been optimized, in order to make their function obvious. Note that for simplicity the checksums have all been read and calculated using a blank screen and border, with (0, 0, 0) in palette location 0, and with no cursor or overlays. Clearly checksums can be generated for any image or display, and this would typically yield different values for each of the three colors.

1.3.3 CALCULATING THEORETICAL CHECKSUM VALUES

The theoretical checksums can be calculated for a given screen size as shown in Appendix A.3.4. Note that with (S)VGA systems, the border (usually black) is passed through the Palette-DAC as normal pixels. Border pixels are consequently used in the hardware checksum calculations, and therefore must be included when calculating the theoretical value. Note also that if the display is interlaced, then the hardware checksum must be accumulated over two frames (fields in this case), as only the odd (or even) lines will have been included during one frame.

The steps that should be followed are:

1 Determine if border present, if so, determine size (left, right, top, bottom).

2 Calculate theoretical checksum values (using border, screen and palette).

Border calculation

The left and right borders (in pixels), and top and bottom borders (in lines) are readily calculated by reading the relevant timing registers in the (S)VGA CRT controller.

These registers are described in the standard VGA specification, consequently these calculations should be portable across different VGA controllers.

To access these CRT indexed registers, the index must be output to address 3D4h, then the register is accessed at address 3D5h.

```
border_left = (((htot + 5 – 1) &
0x01f) – (hbe & 0x01f)) * 8;
```

```
border_right = (hbs – hde) * 8;
```

Where: htot = horizontal total
(CRT index 00h)

hbe = horizontal blank end
(CRT index 03h)

hbs = horizontal blank start
(CRT index 02h)

hde = horizontal display end
(CRT index 01h)

The horizontal registers are in units of 8 pixels. The htot register is 5 less then the real horizontal total (for VGA). Note that the horizontal blank end register is only the lower 5 bits of the register (upper bits are elsewhere) – for convenience only these bits have been considered here, although strictly speaking the upper bits should have been included.

```
border_top = (vtot + 2 – 1) – vbe;
border_bot = vbs – vde;
```

Where: vtot = vertical total (CRT index 06h)

vbe = vertical blank end (CRT index 16h)

vbs = vertical blank start (CRT index 15h)

vde = vertical display end (CRT index 12h)

Note that the vtot register is 2 less than the actual vertical total number of lines (for VGA).

Calculated checksum examples

For convenience, the following checksum examples have been calculated using a black and then a white screen, and with the border color set to black. This means that the checksums for all three colors are the same.

640×480, 8bpp

left border = 8 pixels, right border = 8 pixels
top border = 8 lines, bottom border = 8 lines

So the theoretically calculated checksum must be calculated for a screen size of 656 by 496.

Checksum = 2CC153h (black, RGB)
Checksum = 84E438h (white, RGB).

800×600, 8bpp

left border = 8 pixels, right border = 8 pixels
top border = 4 lines, bottom border = 4 lines

So the theoretically calculated checksum must be calculated for a screen size of 816 by 608.

Checksum = 338ED5h (black, RGB)
Checksum = 3A90B8h (white, RGB).

1024×768, 8bpp

(Note this may be interlaced on some graphics cards)

left border = 0 pixels, right border = 0 pixels
top border = 0 lines, bottom border = 0 lines

So the theoretically calculated checksum must be calculated for a screen size of 1024 by 768.

Checksum = 442B24h (black, RGB)
Checksum = 0CDBC1h (white, RGB).

1.3.4 READING THE HARDWARE CHECKSUMS FROM THE IMS G174

The steps for this are:

1 Wait for frame flyback

2 Reset the checksum registers (in XGA mode)

3 Wait for frame flyback

4 Read the checksums (in XGA mode)

XGA mode

In order to access the checksum registers in the IMS G174, the device must be put into XGA mode. This is easily achieved, and is demonstrated in Appendix A.3.1:

The steps to put the IMS G174 into XGA mode are:

1 Access the XGA enable register by reading the Pixel Mask Register eight times.

2 Put the device into XGA mode by setting the XGA enable bit (bit 2, i.e. write 04h into the XGA enable register).

3 Set bit 7 in the Test Control Register (write 80h into XGA indexed register 80h).

Note step 3 must be done where the VGA controller does not allow read access to 3C7h. This puts the device into a special mode where read/write accesses to 3C8h and 3C9h actually access the registers normally appearing at 3C6h and 3C7h.

After step 3 the XGA index is accessed at address 3C8h, and the XGA data is accessed at address 3C9h.

Resetting the checksum registers

Before the checksums can be read, they must be reset (to FFFFFFh), then allowed to accumulate over a frame.

To reset the checksums to FFFFFFh, as they are read only, a special bit exists (bit 5, XGA index 80h). The procedure is demonstrated in Appendix A.3.6. The checksums must be reset on entering frame flyback, as otherwise they will start accumulating their values. Note that bit 5 is set high, then set low, without altering the state of the register (which will either be 00h or 80h, as mentioned above).

1.3.5 READING THE CHECKSUM REGISTERS

Reading the checksum registers is straightforward, bearing in mind that all accesses must be XGA indexed, and that they should only be read during frame flyback. The procedure is demonstrated in Appendix A.3.7. First the XGA index must be output to the (new) XGA index register, then the appropriate byte can be read back.

Note that this must all be accomplished within frame flyback, as otherwise invalid data will be read, due to the checksum mechanism accumulating during active display.

If due to processor speed and system considerations, it is not possible to read all three checksums during a single frame flyback, then the example code could be divided to reset the checksums, then read a single color per frame flyback. Clearly the screen and palette, cursor etc. must be held constant during this time.

Checksums with interlaced displays

When the display is interlaced, the checksum registers must be allowed to accumulate over two frames (fields).

Also because it may not be possible to determine the field order presented to the Palette-DAC, the checksum mechanism may accumulate the checksum value in a different field order from that used in the theoretical calculation. Clearly the field order will be correct for half the time. The example program shown in Appendix A.3.4 for simplicity does not cater for interlaced displays. Obviously where the fields are identical (such as on an all black or all white screen) then the field order is irrelevent. Two theoretical checksums could be calculated, one for each of the field orders.

6-bit palette operation

Note that in 6-bit palette operation mode the color field written to the palette (6 bits right justified in an 8-bit field) will be presented to the checksum mechanism left justified in an 8-bit field.

1.3.6 VGA MODE

In order to return the device into VGA mode, the XGA enable bit in the XGA enable register must be set low, i.e. 00h must be written into the XGA enable register. This is actually achieved in a similar way to putting the device into XGA mode, except that the mask register must be accessed as an XGA indexed register, and cannot be accessed at address 3C6h as would normally be performed if the part was in VGA mode. 3C6h is of course the original XGA index port in XGA mode. This procedure is demonstrated in Appendix A.3.2.

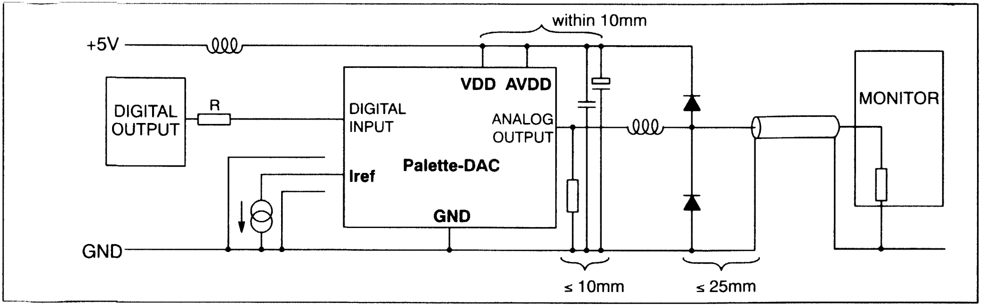
Wait for frame flyback

It is important that the checksum registers are only reset and read during frame flyback. For convenience, a method for synchronizing to frame flyback is shown in Appendix A.3.3. To avoid the possibility of synchronizing to an indeterminate position during frame flyback, it is necessary to first wait for active display, followed by frame flyback. At this point frame flyback will always have just started.

1.4 ANALOG CONSIDERATIONS

1.4.1 SUMMARY

Figure 1.3. Suggested circuit



1.4.2 COMPONENT PLACEMENT

Do keep all Palette-DAC circuitry as close as possible to the output connector to reduce noise pickup and reflections incurred by impedance mismatches. This will also reduce the risk of RFI problems. Keep the DAC termination resistors within 10mm of the DAC outputs to optimize the edge rate performance.

Do keep the current or voltage reference circuitry as close as possible to the Palette-DAC to reduce pickup (within 10mm maximum). For the same reason the Iref or Vref track should not be near any pixel rate signals but should be shielded by GND running either side of it.

1.4.3 POWER SUPPLY REQUIREMENTS

Use VDD and GND power planes to ensure a low inductance supply and reduce RFI emissions.

Use a separated analog VDD plane for the Palette-DAC, connected to the main VDD plane by a ferrite bead (value 0.35μH, impedance about 100Ω at 100MHz). The **VDD** and **AVDD** pins of the Palette-DAC should be connected to this plane. Digital signals should not overlay the analog power plane any more than necessary. To ensure minimum pickup from digital signals, this analog plane can be just under the analog pins of the Palette-DAC and all **VDD** pins.

Use a single 47μF tantalum near to the Palette-DAC for general power plane decoupling, and 0.1μF ceramic chip capacitors immediately adjacent to every pair of **VDD** and **GND** pins.

Do solder the device directly into the PCB to reduce power supply inductance and maximize the effect of the power supply decoupling.

1.4.4 DAC OUTPUTS

DAC termination resistors should be high-accuracy specification and placed within 10mm of the DAC outputs.

Do use double termination as this optimizes the DAC edge rates due to the lower RC time constant and is also more tolerant of impedance mismatches.

The PCB tracks from the termination resistors to the video connector should be sized to form a transmission line of the required impedance. Even so, these traces should still be less than 25mm long to reduce the effect of mismatches and reflections.

Protection diodes to the power rails are recommended at the analog outputs.

1.4.5 DIGITAL INPUTS

Do keep fast digital signals away from sensitive analog circuitry.

The signals most likely to cause problems are the pixel inputs. Do keep the pixel input buffers close to the Palette-DAC to minimize ringing and therefore undershoot and overshoot. If fast drivers are required to achieve timing requirements, a series termination resistor (normally 10–100Ω) helps.

Use the slowest edge rates of signals possible to meet the product datasheet timings. This will cause less RFI and less disturbance of the power supply. Slower speed logic families may often improve overall display quality.

1.4.6 RFI

Keep tracks leading to and from the Palette-DAC short, particularly the pixel address, pixel clock and DAC outputs.

Always use at least a four-layer board with VDD and GND planes. Ensure good VDD and GND decoupling.

Avoid flying leads carrying high frequency signals which are not screened.

Place load resistors as close as possible to the analog outputs.

Use slower speed logic families wherever possible.

1.4.7 REFERENCE CIRCUITRY

Most SGS-THOMSON Palette-DACs can only use Iref reference circuitry. The G173/4C can use internal or external Vref circuitry. For Vref systems an external Vref circuit is recommended for optimum DAC performance.

External reference circuitry must be placed within 10mm of the Palette-DAC. Reference signals must be shielded and not be near digital signals.

The following values apply to Iref and Vref circuits:

Vpeakwhite Vpeakwhite is the peak analog DAC output voltage, including any sync pedestal voltages. On the SGS-THOMSON IMS G17x series of Palette-DACs, sync pedestals are not available and Vpeakwhite will typically be 0.7V.

Rload Rload is the effective load resistance on the analog outputs. If using a doubly terminated 75Ω load, Reffective is 37.5Ω.

Current reference circuit (Figure 1.4**)**

Iref = Vpeakwhite/2.1 × Rload

For Iref systems pin 28 must be connected to 29. An LM334 circuit is recommended (see Fig 1.4) with no decoupling capacitors at all connected to the **Iref** input. If there is excessive noise on the Iref input it is possible to connect a 0.1μF chip capacitor in parallel with a 47μF tantalum capacitor between **Iref** and **AVDD**. High frequency noise in the current reference may be further removed by using a ferrite bead (less than 1μH) in series in the Iref line.

Diode D1 provides temperature compensation - a 1N4148 or equivalent should be used. The ratio of R1 to Rset determines the amount of temperature compensation - typically R1 should be 10Rset, but the optimum ratio should be found by experiment. See the LM334 datasheet for further details.

Voltage reference circuits (Figures 1.5, 1.6)

Rset = 1.235 × 2.1 × Rload / Vpeakwhite

Figure 1.4. Current reference circuit

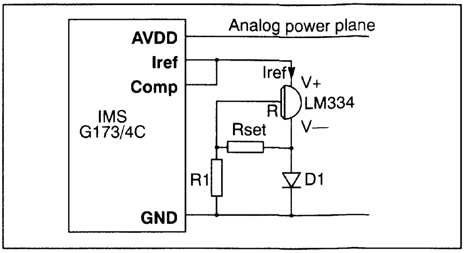


Table 1.4. External Iref components

Component	Value	Description
R1	150Ω	1% resistor
Rset	15Ω	1% resistor
D1	1N4148 or equivalent	diode

Figure 1.5. Using Internal Voltage reference

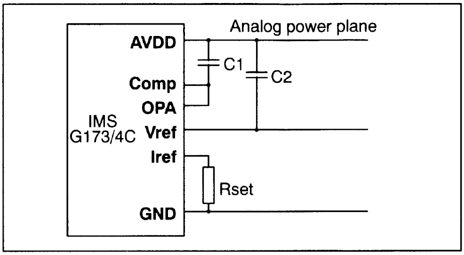


Table 1.5. Internal Vref components

Component	Value	Description
C1–C2	0.1μF	ceramic capacitor
Rset	140Ω (see equation)	1% resistor

Figure 1.6. Using External Voltage reference

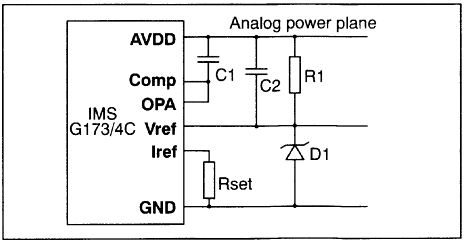


Table 1.6. External Vref components

Component	Value	Description
C1–C2	0.1μF	ceramic capacitor
R1	1KΩ	1% resistor
Rset	140Ω (see equation)	1% resistor
D1	1.2V	voltage reference

A APPENDIX

A.1 EXAMPLE ROUTINE TO SET PIXEL COMMAND AND MASK REGISTERS TO POWER-UP VALUES

Normally after power-up the Pixel Command Register and Mask Register will be 0h and 0FFh respectively. If the contents of these registers is in doubt for any reason then the following routine will ensure that they are both set to their power-up values.

```
; clean start, mask := ff, cmd reg := 0
mov dx, 03c9h
in al, dx            ; reset seq
mov dx, 03c6h
in al, dx
in al, dx
in al, dx
in al, dx
mov al, 0
out dx, al           ; cmd reg := 0
mov dx, 03c9h
in al, dx            ; reset seq
mov dx, 03c6h
mov al, 0ffh
out dx, al           ; mask := ff
```

A.2 EXAMPLE ROUTINE TO IDENTIFY THE PALETTE-DAC

Shown below is an example routine to identify the Palette-DAC, as discussed in Section 1.2.1.

```
; start of dac id example routine
; read cmd reg twice
mov dx, 03c6h
in al, dx
in al, dx
in al, dx
in al, dx            ; music gives id here, #88 / #89
mov bl, al
in al, dx            ; 1st cmd reg read
mov bh, al
in al, dx            ; 2nd cmd reg read
mov cl, al
in al, dx            ; 3rd cmd reg read
in al, dx            ; 4th cmd reg read
in al, dx            ; 5th cmd reg read
in al, dx            ; 6th cmd reg read
mov ch, al

; is it a hi color DAC ?
cmp bh, 0ffh
jnz iddac_hi_color
mov al, not_hi_color_DAC
jmp iddac_exit
  iddac_hi_color:

cmp bl, 0ffh         ; music ?
jz iddac_not_music
mov al, music_hi_color_DAC
jmp iddac_exit
  iddac_not_music:
```

```
cmp bh, cl           ; sierra ?
jnz iddac_not_sierra
cmp bh, ch
jnz iddac_not_sierra
mov al, sierra_hi_color_DAC
jmp iddac_exit
  iddac_not_sierra:
; save palette [1]
mov dx, 03c7h        ; read index
mov al, 1            ; set location index
out dx, al
mov dx, 03c9h        ; data
in al, dx
mov bl, al           ; save r
in al, dx
mov bh, al           ; save g
in al, dx
mov cl, al           ; save b
; palette [1] := ff
mov dx, 03c8h        ; write index
mov al, 1            ; set location index
out dx, al
mov dx, 03c9h        ; data
mov al, 0ffh         ; set to ff, ff, ff
out dx, al
out dx, al
out dx, al
; read palette [1]
mov dx, 03c7h        ; read index
mov al, 1
out dx, al
mov dx, 03c9h        ; data
in al, dx
mov ch, al
; put back contents of palette [1]
mov dx, 03c8h        ; write index
mov al, 1            ; set location index
out dx, al
mov dx, 03c9h        ; data
mov al, bl           ; r
out dx, al
mov al, bh           ; g
out dx, al
mov al, cl           ; b
out dx, al
; hardwired 8not6 pin ?
cmp ch, 03fh
jz iddac_not_hard_wired
mov al, unidentified_DAC   ; 8not6 pin is hardwired to vcc
jmp iddac_exit
iddac_not_hard_wired:
; cmd reg := 2
mov dx, 03c6h
```

```
in al, dx
in al, dx
in al, dx
in al, dx
mov al, 02h          ; set soft 8 bit palette att
out dx, al

; save palette [1]
mov dx, 03c7h        ; read index
mov al, 1            ; set location index
out dx, al
mov dx, 03c9h        ; data
in al, dx
mov bl, al           ; save r
in al, dx
mov bh, al           ; save g
in al, dx
mov cl, al           ; save b

; palette [1] := ff
mov dx, 03c8h        ; write index
mov al, 1            ; set location index
out dx, al
mov dx, 03c9h        ; data
mov al, 0ffh         ; set to ff, ff, ff
out dx, al
out dx, al
out dx, al

; read palette [1]
mov dx, 03c7h        ; read index
mov al, 1
out dx, al
mov dx, 03c9h        ; data
in al, dx
mov ch, al

; put back contents of palette [1]
mov dx, 03c8h        ; write index
mov al, 1            ; set location index
out dx, al
mov dx, 03c9h        ; data
mov al, bl           ; r
out dx, al
mov al, bh           ; g
out dx, al
mov al, cl           ; b
out dx, al

and ch, 0fch         ; att ?
cmp ch, 0fch
jnz iddac_not_att

mov al, att_hi_color_DAC
jmp iddac_exit
  iddac_not_att:

; cmd reg := ff
mov dx, 03c6h
```

```
n al, dx
in al, dx
in al, dx
in al, dx
mov al, 0ffh
out dx, al

; read cmd reg
mov dx, 03c6h
in al, dx
in al, dx
in al, dx
in al, dx
in al, dx
mov ch, al
cmp ch, 0bh          ; acumos ?
jnz iddac_not_acumos
mov al, acumos_hi_color_DAC
jmp iddac_exit
  iddac_not_acumos:

cmp ch, 0ffh
jnz iddac_not_inmos
mov al, inmos_hi_color_DAC
jmp iddac_exit
  iddac_not_inmos:
mov al, unidentified_DAC

iddac_exit:
```

End of example routine.

A.3 EXAMPLE PROGRAMS FOR READING AND CALCULATING CHECKSUMS

A.3.1 SETTING THE G174 INTO XGA MODE

```
void put_g174_into_xga_mode ()
{
  /* assume we are in vga mode */
  asm mov dx, 0x03c7  /* reset seq */
  asm in al, dx
  asm mov dx, 0x03c6
  asm in al, dx       /* 4 reads of mask reg —> pix cmd reg */
  asm in al, dx
  asm in al, dx
  asm in al, dx
  asm in al, dx       /* 4 reads of pix cmd reg —> xga enable reg */
  asm in al, dx
  asm in al, dx
  asm in al, dx
  asm mov al, 4       /* set bit 2 hi for xga mode */
  asm out dx, al
  asm mov al, 0x080
  asm out dx, al
  asm mov dx, 0x03c9
  asm out dx, al
}
```

A.3.2 SETTING THE G174 INTO VGA MODE

```
void put_g174_into_vga_mode ()
{
  /* assume we are in xga mode */
  asm mov dx, 0x03c8  /* xga index reg */
  asm mov al, 0x064   /* mask index */
  asm out dx, al
  asm mov dx, 0x03c9  /* xga data reg */
  asm in al, dx       /* 4 reads of mask reg -> pix cmd reg */
  asm in al, dx
  asm in al, dx
  asm in al, dx
  asm in al, dx       /* 4 reads of pix cmd reg -> xga enable reg */
  asm in al, dx
  asm in al, dx
  asm in al, dx
  asm mov al, 0       /* set bit 2 lo for vga mode */
  asm out dx, al
}
```

A.3.3 WAITING FOR FRAME FLYBACK

```
void wait_for_frame_flyback ()
{
  /* use the vertical retrace when bit vr == 1
   * wait for 0 (video data), then wait for 1 (blanking),
  char bit_vr;
  volatile char byte;
  asm mov dx, 0x03da
  asm in al, dx
  asm mov byte, al
  bit_vr = (byte >> 3) & 1;
  while (bit_vr == 1) /* wait for active display */
  {
    asm mov dx, 0x03da
    asm in al, dx
    asm mov byte, al
    bit_vr = (byte >> 3) & 1;
  }
  asm mov dx, 0x03da
  asm in al, dx
  asm mov byte, al
  bit_vr = (byte >> 3) & 1;
  while (bit_vr == 0) /* wait for frame flyback */
  {
    asm mov dx, 0x03da
    asm in al, dx
    asm mov byte, al
    bit_vr = (byte >> 3) & 1;
  }
}
```

A.3.4 CHECKSUM CALCULATION

```
void calculate_chksums (long int results [3],
                        int screen_width, int screen_height,
                        char palette [256][3])
{
  /* here screen_width and screen_height must include the borders */
  int color, x, y;
  for (color = 0; color < 3; color++) /* reset checksums */
  {
    results [color] = 0x0FFFFFFL;
  }
  for (y = 0; y < screen_height; y++)
  {
    for (x = 0; x < screen_width; x++)
    {
      char pixel;
      if (on_screen)     /* valid pixel */
        pixel = read_pixel (x, y);
      else               /* else border pixel */
        pixel = 0;
      for (color = 0; color < 3; color++)
      {
        results [color] ^= (long) (palette [(int) pixel] [color] << 2);
        results [color] <<= 1;
        if ((results [color] & 0x01000000L) == 0)
        {
          results [color] |= 1;
          results [color] ^= (long) 0x0c20000L;
        }
        results [color] &= (long) 0x0FFFFFFL;
      }
    }
    plot_pixel (0, y, 255); /* progress indicator, plot on line done */
  }
}
```

A.3.5 BORDER CALCULATION

```
/* calculate right hand border */
{
  char hde, hbs;
  asm mov dx, 0x03d4
  asm mov al, 0x02     /* hor blnk start index */
  asm out dx, al
  asm mov dx, 0x03d5
  asm in al, dx
  asm mov hbs, al
  asm mov dx, 0x03d4
  asm mov al, 0x01     /* hor disp end index */
  asm out dx, al
  asm mov dx, 0x03d5
  asm in al, dx
  asm mov hde, al
  border_right = (hbs – hde) * 8;  /* 8 pixels per char */
}
```

```
/* calculate left hand border */
{
  char hbe, htot;
  asm mov dx, 0x03d4
  asm mov al, 0x03       /* hor blnk end index */
  asm out dx, al
  asm mov dx, 0x03d5
  asm in al, dx
  asm mov hbe, al
  asm mov dx, 0x03d4
  asm mov al, 0x00       /* hor tot index */
  asm out dx, al
  asm mov dx, 0x03d5
  asm in al, dx
  asm mov htot, al
  border_left = (((htot + 5 - 1) & 0x01f) - (hbe & 0x01f)) * 8;
}
/* calculate bottom border */
{
  char vde, vbs;
  asm mov dx, 0x03d4
  asm mov al, 0x15       /* vert blnk start index */
  asm out dx, al
  asm mov dx, 0x03d5
  asm in al, dx
  asm mov vbs, al
  asm mov dx, 0x03d4
  asm mov al, 0x12       /* vert disp end index */
  asm out dx, al
  asm mov dx, 0x03d5
  asm in al, dx
  asm mov vde, al
  border_bot = vbs - vde;
}
/* calculate top border */
{
  char vbe, vtot;
  asm mov dx, 0x03d4
  asm mov al, 0x16        /* vert blnk end index */
  asm out dx, al
  asm mov dx, 0x03d5
  asm in al, dx
  asm mov vbe, al
  asm mov dx, 0x03d4
  asm mov al, 0x06        /* vert total index */
  asm out dx, al
  asm mov dx, 0x03d5
  asm in al, dx
  asm mov vtot, al
  border_top = (vtot + 2 - 1) - vbe;
}
```

A.3.6 RESETTING CHECKSUMS

```
void reset_chksums ()
{
  put_g174_into_xga_mode ();
  wait_for_frame_flyback ();
  asm mov dx, 0x03c8
  asm mov al, 0x080   /* index of test reg */
  asm out dx, al
  asm mov dx, 0x03c9
  asm in al, dx       /* read previous value */
  asm or al, 0x020    /* set bit */
  asm out dx, al
  asm and al, 0x0df   /* reset bit */
  asm out dx, al
  put_g174_into_vga_mode ();
}
```

A.3.7 READING CHECKSUMS

```
void read_the_chksums (long int chksums [3])
{
  char red_byte_0,   red_byte_1,   red_byte_2;
  char green_byte_0, green_byte_1, green_byte_2;
  char blue_byte_0,  blue_byte_1,  blue_byte_2;
  put_g174_into_xga_mode ();
  wait_for_frame_flyback ();
  /*wait_for_frame_flyback ();*/ /* interlaced mode only */
  asm mov dx, 0x03c8
  asm mov al, 0x086      /* red byte 0 */
  asm out dx, al
  asm mov dx, 0x03c9
  asm in al, dx
  asm mov red_byte_0, al
  asm mov dx, 0x03c8
  asm mov al, 0x087      /* red byte 1 */
  asm out dx, al
  asm mov dx, 0x03c9
  asm in al, dx
  asm mov red_byte_1, al
  asm mov dx, 0x03c8
  asm mov al, 0x088      /* red byte 2 */
  asm out dx, al
  asm mov dx, 0x03c9
  asm in al, dx
  asm mov red_byte_2, al
  asm mov dx, 0x03c8
  asm mov al, 0x089      /* green byte 0 */
  asm out dx, al
  asm mov dx, 0x03c9
  asm in al, dx
  asm mov green_byte_0, al
  asm mov dx, 0x03c8
  asm mov al, 0x08a      /* green byte 1 */
  asm out dx, al
  asm mov dx, 0x03c9
  asm in al, dx
  asm mov green_byte_1, al
  asm mov dx, 0x03c8
  asm mov al, 0x08b      /* green byte 2 */
  asm out dx, al
  asm mov dx, 0x03c9
  asm in al, dx
  asm mov green_byte_2, al
  asm mov dx, 0x03c8
  asm mov al, 0x08c      /* blue byte 0 */
  asm out dx, al
  asm mov dx, 0x03c9
  asm in al, dx
  asm mov blue_byte_0, al
  asm mov dx, 0x03c8
  asm mov al, 0x08d      /* blue byte 1 */
  asm out dx, al
  asm mov dx, 0x03c9
  asm in al, dx
  asm mov blue_byte_1, al
  asm mov dx, 0x03c8
  asm mov al, 0x08e      /* blue byte 2 */
  asm out dx, al
  asm mov dx, 0x03c9
  asm in al, dx
  asm mov blue_byte_2, al
  chksums [0] = (long) red_byte_0   | (((long) red_byte_1   | ((long)
          red_byte_2   << 8)) << 8);
  chksums [1] = (long) green_byte_0 | (((long) green_byte_1 | ((long)
          green_byte_2 << 8)) << 8);
  chksums [2] = (long) blue_byte_0  | (((long) blue_byte_1  | ((long)
          blue_byte_2  << 8)) << 8);
  put_g174_into_vga_mode ();
}
```

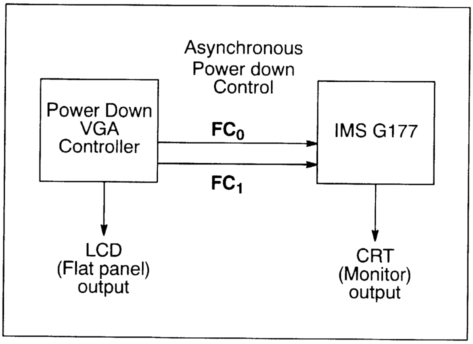
DESIGNING WITH THE IMS G177

2.1 INTRODUCTION

The IMS G177 has been specifically designed for lap top, notebook and all other types of portable PC. The G177 produces a high quality analog output to drive a conventional full color VGA monitor, giving the portable PC a standard and quality of graphics output that is comparable with any high-end VGA desk-top PC. The IMS G177 also boasts the world's lowest power consumption and smallest package for a 6-bit Palette-DAC, down to typically 5µA.

For ease of design the G177 is completely backwards compatible with the SGS-THOMSON IMS G176, the industry standard Palette-DAC for VGA graphics. The power-down features of the G177 are controlled by two Feature Control pins (**FC[0]** and **FC[1]**) which operate asynchronously. The two pins are simply connected directly to the VGA Controller without any extra hardware or software overheads.

Figure 2.1. Ease of interfacing the IMS G177



2.2 INTERFACING TO THE VGA CHIPSET

2.3 INTERFACING THE IMS G177 TO THE CIRRUS LOGIC 610/620

The Cirrus Logic VGA chip set has two bits in the Feature Control Register which can be assigned as general purpose outputs. The Cirrus Logic Power Down VGA chip set ('C' revision) has a standard BIOS which defines these pins to be directly compatible with the G177. The two pins on the 'B' revision chip set are programmable and can be initiated to be fully compatible with the G177. In such a configuration, the **FC[1:0]** outputs on pins 4, 5 of the CL-GD610-C (and pins 20, 21 of the standard CL-GD610) can be connected directly to the **FC[1:0]** inputs of the IMS G177 (pins 15, 17).

The G177 and CL-610/620-C solution provides a very low power implementation while providing an extremely high quality combination of graphics outputs.

2.4 INTERFACING THE IMS G177 TO THE CHIPS AND TECHNOLOGIES 82C455/6

Two pins on the Chips and Technologies chip set (called **ERMEN/** and **TRAP/**) can be redefined as general purpose outputs. It is suggested by Chips and Technologies that **TRAP/** be redefined as an **CRT/$\overline{\mathbf{LCD}}$** information strobe, although the sense would need to be inverted from that normally provided to connect to the **FC[0]** pin of the G177. Therefore to directly connect the 82C455/6 the G177 it is recommended that:

ERMEN/ be redefined as $\overline{\mathbf{PWRDN[2]}}$
TRAP/ be redefined as **LCD/$\overline{\mathbf{CRT}}$**
ERMEN/ = **FC[1]**
TRAP/ = **FC[0]**

with re-definitions programmed when the 82C455/6 is initialized.

There are other solutions open to the designer. If the user requires that **ERMEN/** and **TRAP/** remain undefined or at their default use, then some of the *inputs* to the 82C455/6 can be used instead. The 82C455/6 enter different power management states according to the state of two input pins (or one pin, depending on the device in question). The interpretation of these pins is not programmable but they can be tapped off as inputs to the IMS G177.

82C456 Two Available States

With direct connections of:

VDD = **FC[1]**
PWRDN[2] = **FC[0]**

the G177 will enter REVIEW mode when the 82C456 enters Retire (fully off) mode.

82C455 Three Available States

With the addition of an inverter to provide the state of $\overline{\mathbf{PWRDN[2]}}$ you can connect:

$\overline{\mathbf{PWRDN[2]}}$ = $\overline{\mathbf{FC[1]}}$
PWRDN[1] = **FC[0]**

Under these conditions the G177 will correctly interpret the Chips and Technologies NORMAL, RELAX and RETIRE modes as the G177's NORMAL, REVIEW and STANDBY modes respectively.

2.5 REPLACING OTHER PALETTE-DACS

The SGS-THOMSON IMS G177 can replace the Brooktree Bt475, AMD Am81EC176 and other Palette-DACs to provide an elegant, high quality solution with greatly reduced power consumption and other features especially tailored for portable PCs.

2.6 REPLACING THE AMD Am81EC176 WITH THE SGS-THOMSON IMS G177

The G177 is a pin for pin replacement for the AMD Am81EC176 and does not require any redesign effort. However, the G177 consumes significantly less power than the AMD device and also includes integrated comparators for space saving and cost reduction.

The connections made are:

VDD=FC[1]
SLEEP=FC[0]

2.7 REPLACING THE BROOKTREE Bt475 WITH THE SGS-THOMSON IMS G177

The Brooktree device requires a software overhead to access the control register to control its power management function. This is not a preferred option for the IMS G177 as direct connection to the hardware leads to a simpler interface between the Palette-DAC and the rest of the system hardware and software. Powering down of screens and control of the IMS G177 can be achieved without software action by the use of the asynchronous Feature Control (FC) input pins. The VGA controllers take in and/or provide pins which can be used directly for the G177, giving effortless asynchronous operation for all modes. This is described in more detail in the previous sections.

The IMS G177 consumes significantly less power than the Bt475, does not require additional external circuitry to turn the current reference off during sleep mode and occupies around half the area of the Brooktree package.

2.8 REPLACING THE AVASEM AV36L77 WITH THE SGS-THOMSON IMS G177

The IMS G177 is pin and function compatible with the AV36L77. Using the G177 is therefore simply a matter of substitution, and provides a reduction in power consumption from typically 10mA to 130μA or from 1mA to 5μA depending on the standby mode used.

CVC APPLICATION NOTES

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

32-BIT COLOR VIDEO CONTROLLER UPGRADE COMPATIBILITY: A DESIGN GUIDE

GRAPHICS APPLICATIONS GROUP, BRISTOL

3.1 INTRODUCTION

This Application Note describes how designers can integrate products from the SGS-THOMSON Color Video Controller (CVC) family, into raster scan based video display systems featuring pixel rates up to 135MHz. The features, implementation and the conformance of compatibility of the IMS G332 and the IMS G335 devices are included.

The IMS G332 controller provides an on-chip video RAM manager allowing simple configuration of the framestore designed for 110MHz pixel frequency, supporting 16-bit pixels respectively, non-interlaced with a resolution of 1280×1024 at 60Hz. In order to display 16-bit pixels with “flicker-free” frame rates, typically 72Hz, at this resolution the IMS G335 device features an enhanced pixel sampling scheme allowing application of off-the-shelf TTL logic. The improved controller also features split-SAM support, 24-bit color modes, and new XGA compatible pixel and cursor formats.

The note is written as a designers guide. It discusses the various hardware interfaces of the CVC devices, suggesting typical hardware support implementations. Additional proposals on maintaining IMS G335 enhancement compatibility are discussed with particular relevance to designers requiring direct, pin compatible, IMS G332 to G335 upgrade compatibility of their graphics system.

3.2 FEATURES OF THE COLOR VIDEO CONTROLLER

3.2.1 OVERVIEW

The Color Video Controller devices are designed to support graphics systems based on dual-ported video RAM architecture. The controller architecture allows a reduction in access requirements to the serial port by sampling multiple pixels in a single clock period. These pixels are then serialized to the full dot rate and presented, via a look-up table RAM, to the three high speed DACs. All CVCs feature a programmable Video Timing Generator (VTG) which allows a very wide range of monitors to be driven at different resolutions and frame rates.

An important feature of CVC architecture is that they are designed to employ *seamless midline updates*; the device performing video RAM shift register updates during the process of displaying a line. This is particularly useful when the processor accessing the framestore does so in a sequential byte access pattern throughout the defined memory region. The CVC's seamless midline update makes the screen resolution independent of the hardware architecture of the framestore, allowing all video RAM to be addressed sequentially.

This provides a high level of flexibility to the user's allocation of memory, since contiguous video RAM usage is supported with any arbitrary resolution. If the framestore in a graphics system consists of 1MByte of video RAM, the CVC architecture will allow this memory to be used at different resolutions; for example 640×480 with 16bpp, 800×600 or1024×768 with 8bpp, or 1280×1024 with 4bpp.

The IMS G332 and G335 CVC devices exhibit the following features:

- Packed pixel latching and serialization to the full video rate on-chip allowing data from the video RAM serial port to be shifted out at *dot frequency*/4, *dot frequency*/8, *dot frequency*/16 or *dot frequency*/32 .
- A range of pixel depths, programmable via a single register.
- All controllers integrate a 64×64×2-bit hardware cursor on chip.
- Fully programmable VTG supporting a wide range of standard synchronization formats.
- Programmable Framestore Manager with *seamless midline update* ability.
- On-chip Phase Lock Loop (PLL) technology allowing programmable high frequency internal clock rates to be sourced from a single crystal oscillator.
- Three 256 location 8-bit color palettes with 8-bit high speed DACs.
- All non pseudo color modes are gamma corrected via the look-up table.
- A general purpose micro port
- Master and slave modes of operation.

3.2.2 SUMMARY OF IMS G335 ENHANCED FEATURES

The IMS G335 Color Video Controller features a number of system level improvements allowing flexible, higher performance graphics controller systems to be designed more easily. The IMS G332 and G335 devices have been designed to be software and hardware compatible. The new device is a drop-in replacement as the pins used for enhanced functionality were previously defined as **HoldToGnd**. The IMS G335 device is available in the same speed options as the IMS G332 with the addition of 135MHz.

- **Maximum pixel rate**: The IMS G335 has been designed to run at 135MHz pixel frequency. This allows it to support resolutions of up to 1280×1024 at 75Hz.
- **Support for split-SAM video RAMs**: To provide support for video RAMs with split-SAM capability a new mode has been introduced, selectable through Control Register B. Various pins have been re-assigned to support new split-SAM functions on the IMS G335 device.
- **External pixel sampling mode**: A new externally-strobed sampling mode has been implemented to allow higher sampling speeds to be achieved through shorter data set-up and hold times, greatly simplifying high speed system design. Backwards compatibility with the IMS G332 is provided through support of the existing sampling mode, selectable through Control Register B.
- **Shorter DMA transfer cycles during line flyback**: On the IMS G332, DMA transfer cycles that straddle two visible picture lines are rescheduled to the start of the backporch of the second line, (instead of holding the bus during frontporch+sync+backporch) thus reducing the time that bus access is denied to other devices. The IMS G335 architecture further improves this by rescheduling to the point where the total DMA time is equal to the transfer delay. This reduces the bus control time of the device to an absolute minimum, giving maximum bandwidth for processor accesses to memory.
- **Additional synchronization waveform mode**: Recent multisync monitors specify continuous line sync signals during frame flyback to maintain stability of the monitor PLL. This mode (not on the IMS G332) is available on the IMS G335 device.
- **Improved micro port timing**: Micro port control timings on the IMS G335 device are independent of the system clock, although total read/write cycle times are still system frequency dependent (but are reduced in value from the IMS G332).
- **24-bit color support for the IMS G335**: 24-bit color modes have been designed into the architecture of the IMS G335 using smaller multiplex ratios in the packed pixel sampling of data applied at the pixel port of the controller. The smaller multiplex ratios of 2:1 in interleaved framestore designs and 1:1 in non-interleaved designs dictate maximum video rates related to the maximum pixel sampling rate at the video RAM serial port. Today's video RAM technology permits sampling rates which allows the IMS G335, with an interleaved framestore, to support 24bpp images with a dot rate of almost 50MHz.
- **Support for XGA multimedia pixel format**: The XGA graphics standard from IBM supports a 16-bit true color format, split 5,6,5 between red, green and blue fields respectively. This mode, targetted towards cost-sensitive multimedia applications, is supported by the IMS G335.
- **Complement cursor and XGA cursor options**: An option has been added to the IMS G335 to replace one of the cursor colors with pixel complement. Instead of a cursor palette color, the complement of the background color bit value is generated and displayed, ensuring that the cursor pixels programmed in this way are always visible. This also allows XGA compatible cursor mapping to be selected through Control Register B.
- **Even field interlace mode**: A new interlace mode has been added, permitting displayed fields to have an even number of lines. This allows support of double scan modes, similar to those used in the VGA standard.

3.3 THE COLOR VIDEO CONTROLLER PROCESSOR INTERFACE

All CVCs have a bidirectional 24-bit processor interface consisting of a multiplexed address and data bus and several control signals, through which the VTG screen description registers, color and cursor look-up tables, cursor store and other registers are programmed. The CVC microport is designed as a memory mapped peripheral with standard control signals; **ReadnotWrite**, **notOutputEnable**, **notChipSelect** and **Wait**, which allow an interface to be defined with any processor.

Table 3.1. Micro port control signals

Signal	I/O	Function	Implementation notes
notChipSelect	I	The falling edge of this signal latches the address presented on the multiplexed address/ data bus and initiates the sampling of **ReadnotWrite** 1 SClk period later. In IMS G335 designs the falling edge of **notChipSelect** latches the address from the **ADBus**.	The control of this signal allows the interface to the IMS G335 to be asynchronous. This is achieved by preventing the generation of this signal between successive accesses to the CVC.
ReadnotWrite	I	If this signal is low shortly after **notChipSelect** goes low, the cycle is a write; if it is high the cycle is a read. Additionally on a write cycle, the rising edge of **ReadnotWrite** strobes in the data.	This signal is sampled 1SClk after **notChipSelect** is sampled active (low), with a 10ns setup time.
notOutputEnable	I	This signal is used only during read cycles, and enables the read data onto the **ADBus**. It should be kept high at all other times.	
Wait	O	Wait eliminates the need for an external wait state generator. It is driven high shortly after **notChipSelect** goes low, and is driven back low when the G332 is ready for write data to be strobed by **ReadnotWrite** or read data is about to become valid on the **ADBus**.	This signal supports a more asynchronous interface (with respect to the video clock internal to the CVC). External interface logic would typically sample the falling edge of **Wait** to terminate the current cycle.

3.3.1 ADDITIONAL MICRO PORT FEATURES

Word alignment: All CVCs are designed for use with 32 and 64-bit processors and therefore support both 32 and 64-bit word alignment. With 32-bit word alignment selected, the least significant address bit is **ADBus2**; with 64-bit word alignment selected on **ADBus3**. This applies only on host processor accesses to the micro port, in which a 10-bit address is input to the device, and not on DMA transfer cycles in which a 22-bit word address, **ADBus2–23** is generated by the framestore manager.

DMA transfer operation: Two signals **BusRequest** and **TransferA**[1] (or B or both A and B), are used to control the synchronous reloading of the video RAM shift register. To retain the minimum 4:1 packed pixel format, all CVCs have an interleaved memory mode requiring two transfer signals **TransferA** and **TransferB**. The CVC asserts **BusRequest** to obtain use of the **ADBus** to issue a transfer address to the video RAM address port. In IMS G332 designs a **BusGranted** triggers the assertion of the transfer signal(s), allowing the external generation of RAS and CAS to the video RAMs. The point at which the transfer signal returns low, performing a re-load of the shift registers, is critical since mid-line updates must be seamless.

With the IMS G335 split-transfer mode the point at which the transfer is timed to take place need not be synchronous to **notShiftClk**. In this mode the designer is free to specify the point of transfer, inherently providing additional benefits of improved processor bus bandwidth and less critical timings in placing the transfer point.

Video RAM address increment: This feature aids the system designer in optimizing the use of external logic components performing row and column address multiplexing and allows greater efficiency of framestore usage when interlaced displays are required.

A common scenario in the allocation of row and column addresses is how to achieve simplicity and minimization of multiplexing logic for both proces-

[1] The equivalent signals to **TransferA/B** on the G332 are **DTA/B** on the G335.

sor accesses to the framestore and CVC presentation of the transfer address during DMA cycles. To ensure that the processor accesses the internal structure of the memory devices comprising the framestore correctly, the designer must allocate high order addresses on the falling edge of $\overline{RAS}$, such that the drawing processor sequentially accesses the memory with column addresses least significant corresponding to the structure of the screen. Remember, the CVC will perform row transfers and shift pixels column by column. So it is critical to ensure that the processor accesses the framestore in the same manner.

When the CVC outputs a 24-bit transfer address during DMA cycles, the count sequence of this address is defined in Control Register A.

Table 3.2. Video RAM address increment as set by Control Register A, bits 12 and 13

Register Bit		Non-interlace	Interlace	
13	12	Increment	Increment	Second field offset
0	0	1	not used	
0	1	256	2	1
1	0	512	512	256
1	1	1024	1024	512

The value selected directly relates to the number of row and column address bits allocated to the framestore multiplexed address bus, as defined in the processor to framestore interface. The address increment allows row and column address latches used in the processor interface to be suitable for the CVC interface. Table 3.3 shows the address and increment assignments for the IMS G332 and G335 CVCs.

Table 3.3. Video RAM address and increment assignment for IMS G332 and G335

IMS G332/G335 framestore architecture	RA0–8	CA0–7	Increment
128K×32 non-interleaved	A10–18	A2–9	256
128K×32 interleaved	A11–19	A3–10	512

IMS G332/G335 framestore architecture	RA0–8	CA0–8	Increment
256K×32 non-interleaved	A11–19	A2–10	512
256K×32 interleaved	A12–20	A3–11	1024

Note the absence of address bit A2 in interleaved framestore assignments. This bit decodes which bank is selected during processor accesses and is ignored during CVC transfer cycles.

The designer will appreciate that the address increments are provided to optimize row and column address latch and multiplexing logic used by both the processor and the CVC during transfer cycles.

The control of the address increment directly allows efficient use of framestore memory in interlaced displays, so that successive fields contain pixel data throughout the same shift register. The address increment can be used to provide odd and even field offsets as SAM start addresses, maximizing the functionality of the video RAM architecture in interlaced display modes. The framestore manager will typically use half the increment defined for non-interlaced displays such that the second field displays pixels from half way through the SAM, corresponding to the most significant bit of the column address field.

3.3.2 GENERATION OF THE COLOR VIDEO CONTROLLER MICRO PORT STROBES

From the IMS G332 micro port read and write cycle timings it follows that the interface logic producing the micro port control strobes must be derived using synchronous techniques clocked from **notSerialClk**. Failure to design a robust synchronous interface between the processor and the CVC may result in problems at low video frequencies where the period of the **notSerialClk** is large compared to the processor clock.

The IMS G332 does, however, support asynchronous design approaches by asserting the **Wait** signal up to 25ns after **notChipSelect** was asserted low. The **Wait** signal is asserted high for up to 3**SClk**s and 4**SClk**s in write and read cycles respectively in which the overall cycle time is 4**SClk**s and 7**SClk**s, respectively.

The IMS G335 micro port timing specification reduces the processor interface's dependency upon the **notSerialClk** frequency, allowing the micro port strobes to be asynchronously generated within an active write or read cycle. Again the **Wait** signal output by this controller simplifies the interface logic operating over a wide range of video frequencies.

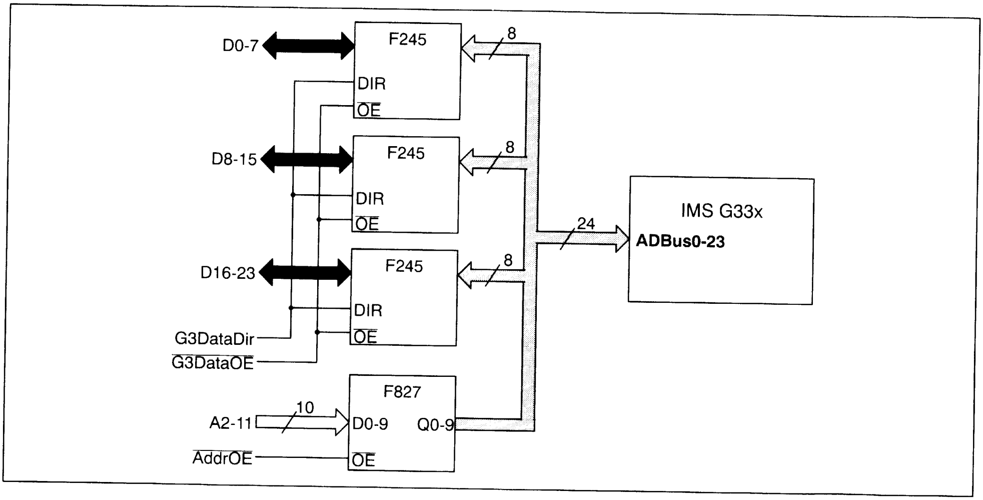
A timing dependency upon the CVC's internal video clock, when generating signals to interface to the micro port is allowing data transferred through the port to be introduced correctly into the CVC's pixel pipeline. This criteria applies to all high speed pipelined graphics devices, and in CVC designs this time is always related to SClk periods. However, the external micro port interface control logic should be designed to prevent the generation of successive **notChipSelect** signals using a conditional input which guarantees 2 SClk periods have expired after triggering on the rising edge of **notChipSelect**.

3.3.3 PROCESSOR INTERFACING TO THE CVC MICRO PORT

To minimize pin-out requirements, the CVC micro port supports a 24-bit multiplexed address/data bus allowing efficient read/write access to the triple 8-bit palettes, the cursor RAM and all 24-bit registers in the device's memory map with no overhead in additional address pins. During processor accesses a 10-bit address must be presented with the specified setup and hold time on the falling edge of **notChipSelect**. In a 32-bit system the CVC's internal register set is addressed using **ADBus2-11**.

Most microprocessors available today provide a separate address and data bus. In general this allows higher bandwidth, but also provides flexibility in more complex external memory interface designs which may use address pipelining, write posting and other memory interface methodologies to enhance the processor's memory addressing and data access requirements. Additional logic is required to allow such processors to interface with the CVC micro port. A typical example is shown in Figure 3.1.

Figure 3.1. Typical micro port interface logic implementation



There are a number of points to note about this implementation:

1 The **G3DataDir** strobe is derived from the processor's R/W̄ signal(s) asserted during a decoded access to the controller.

2 **G3DataOE** and **AddrOE** control the multiplexing of the address and data to the CVC micro port. Note also the address buffer enables the lower memory address bits only.

3 When the framestore manager initiates a transfer cycle **G3DataOE** and **AddrOE** are all high since the **ADBus** will present a 22-bit transfer address on **ADBus2–23**.

4 A Programmable Logic Device (PLD) typically controls the OE and DIR inputs to the transceiver logic. In addition to controlling the flow of data, state sequences clocked from **notSerialClk** are used to generate **notChipSelect**, **ReadnotWrite** and **notOutputEnable**. Another function, which should be included in the PLD, is timing of **notSerialClks** between accesses to avoid violation of the tCLCL parameter.

3.4 FRAMESTORE MANAGEMENT

The CVC devices are designed to sample and serialize between 1 and 32 packed pixels in a single shift clock period. In order to provide support for 24bpp sampling on the pixel port of the IMS G335, the internal clocking and serialization methodology was changed (from a *4:1* minimum serialization) to allow *2:1* or *1:1* serialization. Typically however, the *4:1* serialization allows the shift clock rate to be at least a quarter of the required video frequency. This is typical of a simple IMS G332 based graphics system displaying 8-bit pixels. The flexible pixel multiplexing also allows *8:1*, *16:1*, *32:1* pixel multiplexing ratios. The rate of data sampling is increased with pixel depth (in bits), to a point such that, because of the very high video rates supported, it is not possible in some cases to supply pixel data fast enough from a single bank of video RAM.

If the IMS G332 or G335 were required to produce 15/16bpp pixels only two pixels are latched at the pixel port, and thus the highest device rated speed of 110MHz would require 18ns serial port cycle time. The latest technology limits video RAM operating speed to 40MHz defining a serial port cycle time of 25ns. For dot rates greater than 80MHz in these modes, support for additional multiplexing is required.

The IMS G332/G335 provide an interleaved mode to support the multiplexing of true (15/16 bits-per-pixel) color pixels respectively. The framestore must be designed in two interleaved banks which are independently clocked such that pixel data can be latched from each bank separately. Thus half of the serial port cycle time is pipelined.

In interleaved mode, two shift clocks and transfer signals are used to control each bank, the shift clocks effectively running in anti-phase. With the IMS G332 and the IMS G335 devices in real-time transfer mode, **notShiftClkA** with **TransferA**, and **notShiftClkB** with **TransferB** control the respective interleaved memory banks **A** and **B**. On DMA real-time transfer cycles both banks have their shift registers reloaded, which means that these cycles are required at half the frequency compared with non-interleaved mode. The tables below illustrate which pixel resolutions are accommodated in either non-interleaved or interleaved mode.

Table 3.4. Non-interleaved pixel modes with the IMS G332/G335

Register bits				Bits per pixel	ShiftClk period	MUX ratio	Use of LUT
22	21	20	18				
1	1	0	0	24	1 SClk	1:1	Gamma corrected full color *IMS G335 only*
0	1	1	0	8	1 SClk	4:1	Pseudo color
0	1	0	0	4	2 SClk	8:1	Pseudo color
0	0	1	0	2	4 SClk	16:1	Pseudo color
0	0	0	0	1	8 SClk	32:1	Pseudo color

Table 3.5. Interleaved pixel modes with the IMS G332/G335

Register bits				Bits per pixel	ShiftClk period	MUX ratio	Use of LUT
22	21	20	18				
1	1	0	1	24	1 SClk	1:1	Gamma corrected full color *IMS G335 only*
1	0	1	1	16	1 SClk	2:1	Gamma corrected true color
1	0	0	1	15	1 SClk	2:1	Gamma corrected true color
0	1	1	1	8	2 SClk	4:1	Pseudo color
0	1	0	1	4	4 SClk	8:1	Pseudo color
0	0	1	1	2	8 SClk	16:1	Pseudo color
0	0	0	1	1	16 SClk	32:1	Pseudo color

3.4.1 PIXEL SAMPLING WITH THE IMS G332

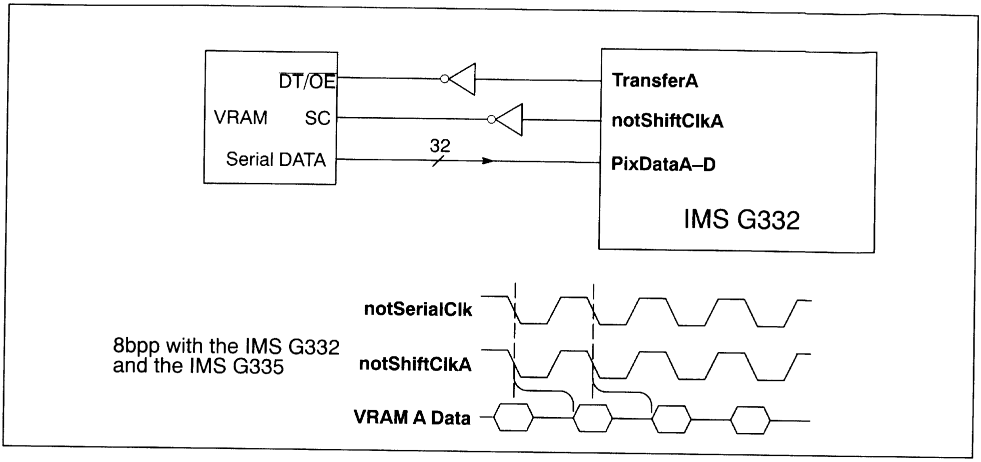
Non-interleaved mode screen transfer operation

The IMS G332 provides two software programmable strobes, **BusRequest** and **TransferA**, which enable them to perform the necessary screen data-transfer cycles on video RAMs to reload their internal shift registers with new data. These may be synchronous updates which happen part way across a line or updates occurring during flyback.

The strobes are controlled by values loaded into the framestore description registers, MemInit and TransferDelay. These are loaded and decremented continuously in turn whenever the display is active (i.e. they count displayed pixel groups).

From the start of any frame, MemInit is loaded first then TransferDelay. An update cycle is requested at the expiry of MemInit and is completed at the expiry of TransferDelay as shown in Figure 3.3. The IMS G332 also outputs a transfer address specifying the new row of pixels to be displayed. It is left to the user to generate RAS, CAS and any other strobes needed from **BusRequest**, **Transfer**, **notSerialClk** and **notShiftClk**.

Figure 3.2. Non-interleaved video RAM



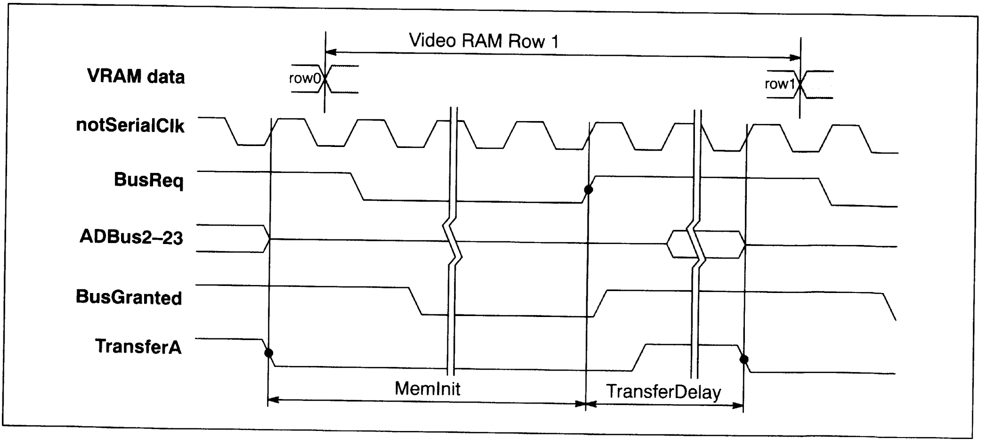
The transfer sequence is as follows:

1 The IMS G332 asserts **BusReq**. When **BusGranted** is given, it drives **TransferA** high and also drives the reload address onto **ADbus23-2**.

2 The IMS G332 waits until the TransferDelay period expires, during which time the system must perform a video RAM access.

3 After a fixed time (TransferDelay Screen Units) relative to the rising edge of **BusReq**, the IMS G332 drives **TransferA** low and tri-states the **ADbus**. One SClk cycle later, **BusReq** will go low.

Figure 3.3. Data transfer sequence



At the start of active display the IMS G332 will begin to count **notSerialClk** cycles and will initiate a further transfer cycle after MemInit cycles of **notSerialClk**. It will then follow the transfer sequence as described above. The falling edge of **TransferA** is synchronized to **notShiftClkA** to allow synchronous mid-line updates. The whole Meminit - TransferDelay sequence is repeated as many times as necessary until the bottom of screen is reached. When the system reaches the end of a scan line any unused pixels in the video RAM shift register are held over to be displayed on the following line.

Thus the period of row transfer operations is MemInit + TransferDelay (displayed pixel groups), and apart from the restrictions quoted in Table 4, it need bear no relation to the screen line length at all. This permits any display line length with any type of video RAM.

Table 3.6. Restrictions on Framestore description parameters

Display	=	Multiple of **notShiftClk** period in SUs
TransferDelay	<	Backporch - 2
TransferDelay	≥	System DMA Latency+ Video RAM Access+1 SClk
Additionally, in an interlaced system only:		
MemInit + Transfer-Delay	=	Display
TransferDelay	<	ShortDisplay

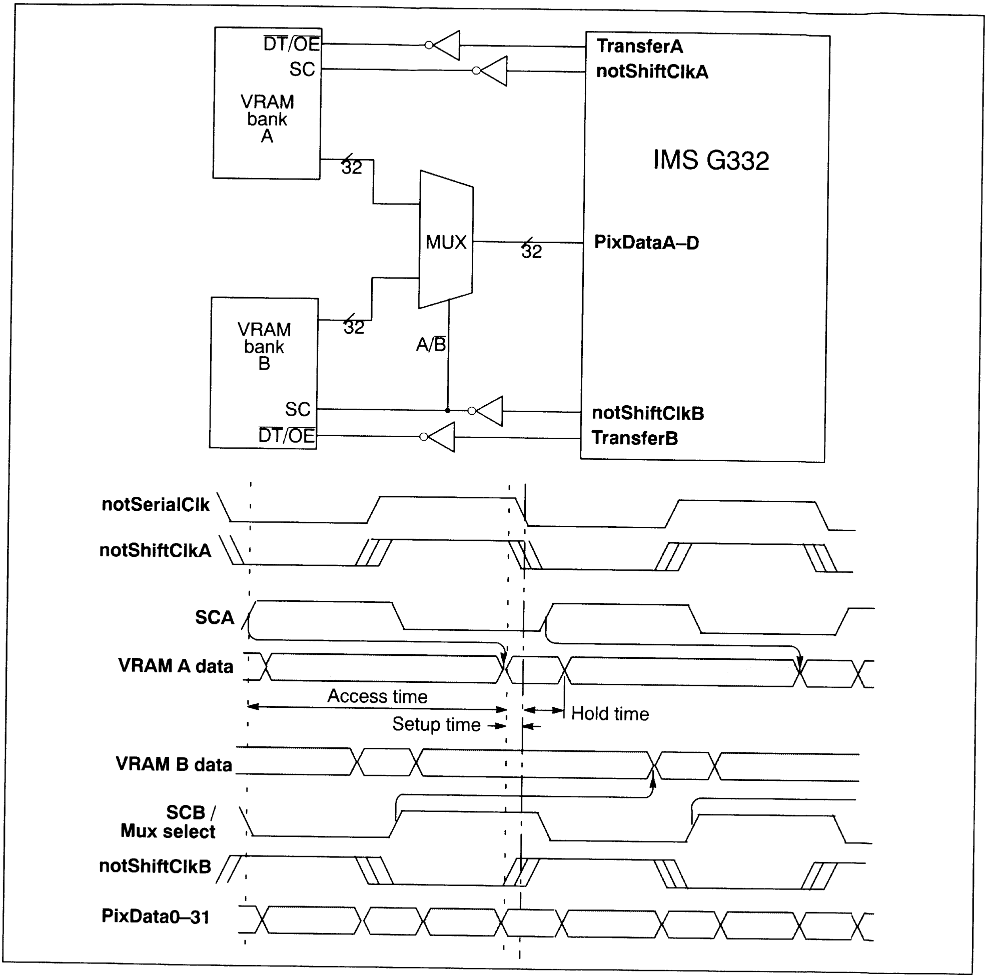
The critical parameter as far as DMA accesses are concerned is TransferDelay which needs to be long enough to allow for the worse case DMA latency of the drawing processor as well as the access time of the video RAMs. With the exception of the first load operation of each frame, the framestore manager only counts **notSerialClk** cycles while the display is active, i.e. it counts displayed pixels.

Interleaved mode

Because of the very high video rates supported by the IMS G332 it is not possible in some situations to supply pixel data fast enough from a single bank of video RAMs. An interleaved mode has been provided to allow two banks of video RAM to be used, each running at half the clock frequency of that required if a single bank were used. 32 bits of pixel data are loaded alternately from one Video RAM bank then the other.

In interleaved mode, **notShiftClkA/B** and **TransferA/B** signals are used to control the two banks, the shift clocks effectively running in anti-phase. The two shift clocks are typically not configured to run in simple antiphase since **notShiftClkA** and **notShiftClkB** differ at the start and end of lines to allow correct multiplexor selection when **notShiftClkB** is used. It can be seen that the falling edge of **notShiftClkA** will cause data to be driven out from Video RAM Bank A into the **PixData** input of the IMS G332.

Figure 3.4. Interleaved sampling mechanism



Interleaved video RAM control: In order to implement an interleaved framestore, the IMS G332 provides a second set of video RAM clock and transfer signals; **notShiftClkB** and **TransferB**. **notShiftClkB** should be setup (enable delayed sampling) as the delayed version of **notShiftClkA** such that the data in each bank is skewed by one half the **notShiftClk** period. The two streams of data can then be multiplexed into the controller. The effect of this is that each bank of video RAM must be clocked at half the rate which would be necessary for a non-interleaved framestore.

An interleaved framestore system is shown in Figure 3.4, together with the **notShiftClk** and video RAM data waveforms for the IMS G332 15/16 bits per pixel format. In these modes, the IMS G332 samples pixel data on both the rising and falling edges of **notSerialClk** (in all other interleaved modes, data is sampled on selected falling edges of **notSerialClk**).

Video RAM reload in an interleaved framestore: The basic reload cycle in interleaved format is the same as the standard format. Because of the skew between **notShiftClkA** and **notShiftClkB**, the **TransferB** strobe must be delayed by a similar amount with respect to **TransferA**. This amount varies depending on the pixel depth, and is automatically adjusted by the IMS G332.

In all cases, the beginning of the DMA cycle is identical to the non-interleaved case, the difference being that the falling edge of **TransferB** is delayed by one half period of **notShiftClkA** (or **B**), with **BusReq** remaining asserted until one period **notSerialClk** after the falling edge of **TransferB**. Also, the reload address driven onto the ADbus remains valid until after the falling edge of **TransferB**.

Use of an interleaved system affects the framestore description parameters in that the effective video RAM shift register length is doubled from that in non-interleaved mode. MemInit must therefore be modified to account for this, leaving TransferDelay the same, so that,

MemInit = (Video RAM serial register length × 2) – TransferDelay (in Screen Units)

3.4.2 IMPROVING THE PIXEL SAMPLING MECHANISM

The IMS G335 devices have an enhanced pixel sampling scheme to allow operation at up to 135MHz, with screen resolutions of 1280×1024 at 75Hz refresh rate. The enhanced pixel sampling scheme, without which the maximum performance of the device cannot be realised in a graphics system, also makes lower performance systems inherently easier to design, compared to that provided on the IMS G332.

The reason for improving the sampling scheme on the IMS G335 is related to the shortest video RAM serial port cycle time of 30ns. This cycle time is required to achieve a 135MHz dot rate using the lowest CVC multiplex ratio of 4:1 in non-interleaved mode and 2:1 in interleaved mode (with a further 2:1 multiplex ratio defined by the external interleaving mechanism). Pixel data is clocked out of the video RAM serial ports by a positive edge on its **SC** pin. This data is then sampled on the **PixData** pins of the G335.

The G335, like the IMS G332, outputs signals (**notShiftClks**) which are designed to be inverted externally in order to provide sufficient buffering to drive the **SC** pins on the required number of video RAMs. The variation in the propagation delay in this external buffering due to capacitive loading from the number of video RAMs being used, type of buffer, operating temperature variations and manufacturing tolerance means that data from the serial port outputs arrives at the pixel inputs with that same tolerance relative to the original clocking signal, **notShiftClk**. If, as on the IMS G332, pixel setup and hold times are specified relative to **notSerialClk**, whose edges are synchronous with those of **notShiftClk**, then for video RAMs operating at 30ns serial cycle time this uncertainty is too large to allow a sufficiently tight sampling window to be specified.

An alternative way of accurately placing the sampling point is designed into the sampling architecture of the IMS G335 family. The **LoadA** and **LoadB** pins, which are derived from the **SC** signals applied to the video RAMs, reset the internal pixel sampling clock, and hence latch the pixel data. Pixel setup and hold times are thus specified with respect to this external input, allowing the propagation delay and tolerance of the **SC** inverter/buffer to be removed from the uncertainty in the sampling point. The IMS G335 specifies a worst case 2.5ns pixel setup and 2.5ns hold relative to **LoadA** and **LoadB**, allowing the propagation delay through the external **SC** inverter/buffer to be removed from the pixel access time. Bit 0 in control register B enables pixel latching using the **LoadA-B** pins, as just described, when set high. If, however, this bit is set low, then the part will operate with the pixel latching method used on the IMS G332.

The improved sampling scheme is not just useful for designing 1280×1024 75Hz systems; the pixel setup and hold time specifications are such that it becomes much easier to design video RAM bank-swapping and interleaving systems for lower frequency implementations too.

Non-interleaved mode single video RAM bank systems

The simplest way of using the **LoadA-B** sampling method is shown in Figure 3.5. The output of the **notShiftClkA** inverter/buffer drives the **SC** pins on the video RAMs, and is additionally fed back directly to the **LoadA** input pin. With this configuration, the only limitation is how much memory can be provided by populating the pixel port with either 256K×4 or 128K×8 video RAMs. Table 3.7 shows how much is available for the G335 with one single bank of each type of video RAM.

Figure 3.5. A single bank of video RAM

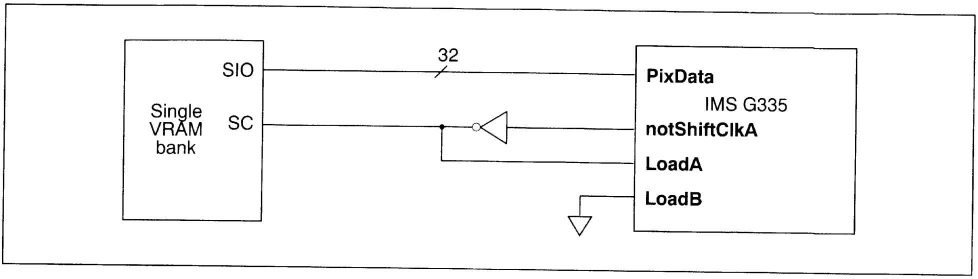


Table 3.7. Framestore size for a single non-interleaved video RAM bank

CVC	128K×8	256K×4
IMS G335	0.5Mbyte	1Mbyte

Non-interleaved mode multiple video RAM bank systems

Multiple banks of video RAMs can be accommodated if care is taken over when the bank swapping is timed to take place. Figure 3.6 shows how two banks can be controlled for a system using 'real-time' (un-split) transfer video RAMs, the principle being easily extended to manage further banks. For example, if the two banks of video RAM were of the 256K architecture, then 10 bits are required to address all 512 rows in both banks. The bottom 9 bits select one row from each video RAM simultaneously and the top bit is used to enable the serial output buffers of either bank. The first D-type latches in the bank select bit at the end of the DMA row transfer cycle. This cannot be directly applied to the video RAM $\overline{\textbf{SOE}}$ pins until the last set of pixels has been sampled from the last row of the previously selected bank. The second D-type is used to synchronize the actual bank swap point appropriately with the pixel sampling points.

Figure 3.6. Two banks of video RAM, using unsplit transfers

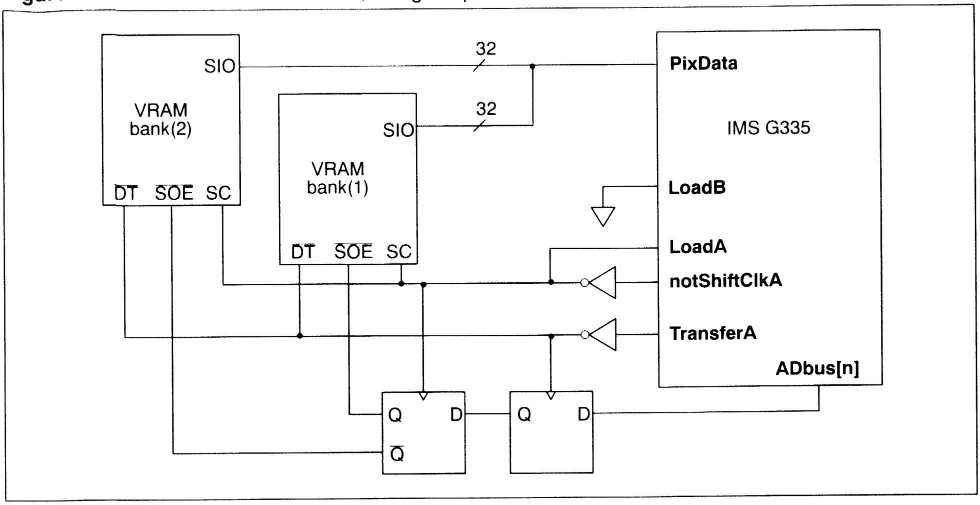


Table 3.8.

Symbol	Description	Min.	Max.	Unit	Notes
tSCC	SC clock period		30	ns	
tPLH	**SOE** low to high time	3.8	8.0	ns	1
tPHL	**SOE** high to low time	3.8	8.0	ns	1
tSEA		–	15	ns	2
tSEZ		3	12	ns	2

NOTES:

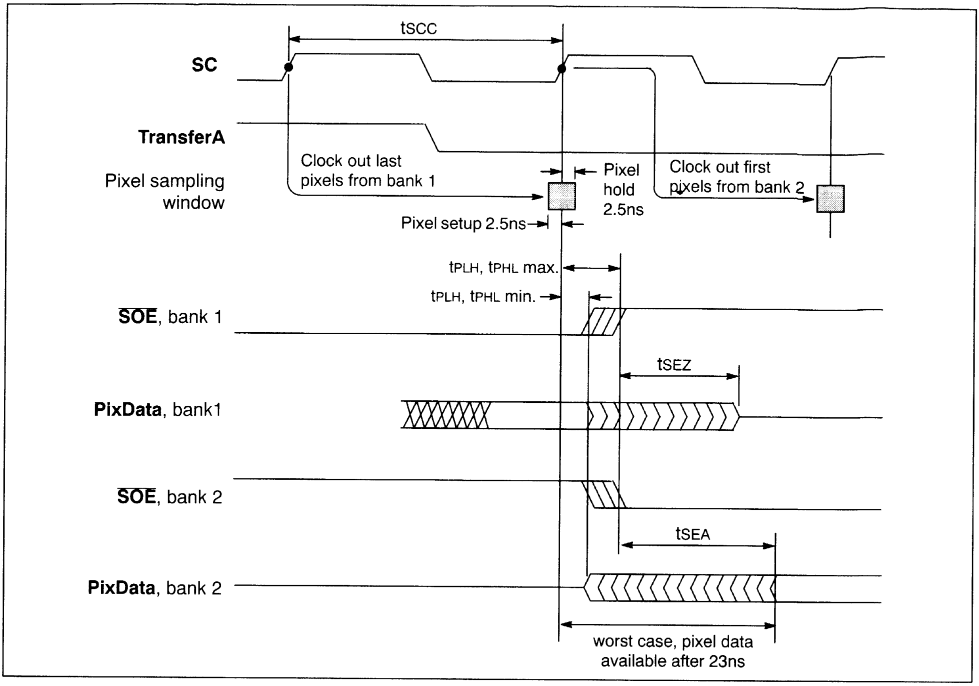
1 These timings apply to 74F74 D-type with tPLH and tPHL from CP → Q,Q̄

2 These timings apply to video RAM type Micron -8 256K x 4, MT42C4256

The pixel hold time for the IMS G335 allows bank swapping to be used with serial clock cycle times of 30ns, as required for operation at 131MHz. Figure 3.7 shows a timing diagram for such a system, with the IMS G335 running 8bpp at 131MHz using two banks of eight -8 256K×4 video RAMs.

An important point to note is that since the SC signal is used to drive **LoadA**, the pixel sampling clock is left active during line flyback, and is only reset after the first edge on **SC** at the beginning of the next line. The same **SC** signal must be used to clock out the new bank select data, rather than using an inverted **notSerialClk** from the IMS G335 as might have been chosen. This is because if a DMA row transfer requiring a bank swap occurred at the end of a line then, if the swap occurred before the sampling clock was reset, the pixel data for the last few pixels on the line would be incorrect (they would actually come from the last row of the new bank). By holding off the bank swap until just before it is required at the beginning of the next line, this potential problem is avoided.

Figure 3.7. Bank swap at 131MHz

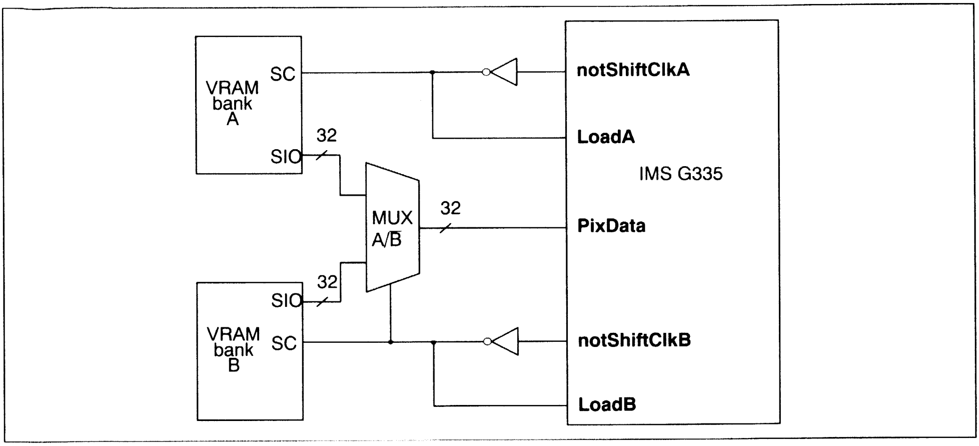


Interleaved mode single video RAM bank systems

The interleaved mode of the IMS G335 is designed to allow higher frequency true color operation.

Figure 3.8 shows a single bank of interleaved video RAMs in which the **notShiftClkB** is used to select which bank is active at a particular instant. Note that **notShiftClkA** cannot be used as the bank select signal since at the end of each display line it stops before **notShiftClkB**, thus resulting in the last few samples being from an incorrectly selected bank.

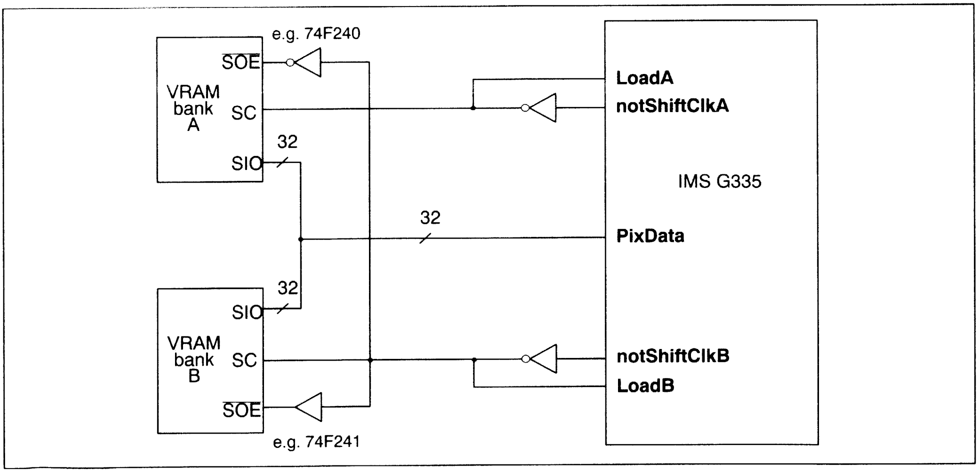
Figure 3.8. A single bank of interleaved video RAMs



The choice of how to implement the bank multiplexer depends on the maximum video dot rate that the system is designed to run at. If the video RAM $\overline{\textbf{SOE}}$ turn on/off timings are sufficiently fast, it is possible to multiplex between the two banks *without using any external multiplexing hardware*. Such a system is shown in Figure 3.9, with its associated timing diagram in Figure 3.10.

Using worst case propagation delays specified for the 74F04 inverters, and with Micron 2Mbit VRAMs (serial output turn on time of 12ns), the system will run at up to 95MHz video rate, allowing a 1280×1024 screen at 8bpp or 1024×768 screen at 16bpp with the IMS G335 at around 85-90Hz frame rate. This is verified by summing tPD (74F04), tSEA (MT42C8256) and the IMS G335 pixel setup time tPVLH. This is 6+12+2.5 = 20.5ns. This means that the VRAM shift clock period must be 41ns minimum. The maximum video rate is then 4/41ns = 95MHz at 16bpp or 47.5MHz at 24bpp using 2:1 interleaving.

Figure 3.9. Interleaving using video RAM $\overline{\text{SOE}}$



Consideration should also be given to the IDT74FCT805A/806A devices which provide 5:1 reduced input capacitance and guaranteed skew between outputs of the same package *and* between packages. These devices are particularly useful as inverters for the **notShiftClk** signals from the CVCs since they minimize the capacitive load presented to the device pins, improving picture quality.

Figure 3.10. $\overline{\text{SOE}}$ interleaved timing diagram for 95MHz 16-bit color screen refresh using the IMS G335

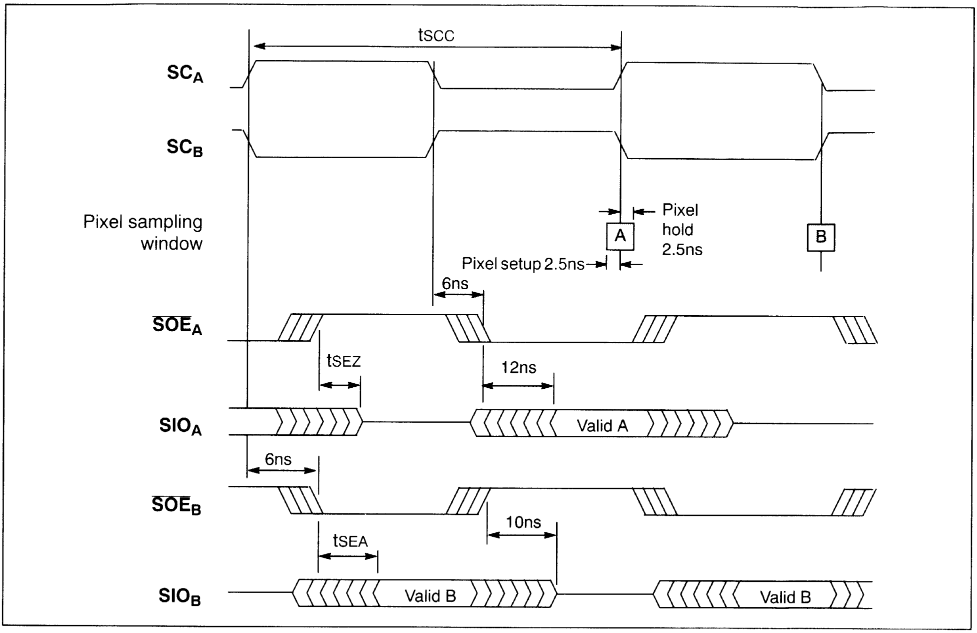


Table 3.9.

Symbol	Description	Min.	Max.	Unit	Notes
tSCC	**SC** clock		41	ns	
tSEA	Access time from Serial Enable (SE)	–	12	ns	1
tSEZ	Serial output buffer turn-off delay from SE	3	10	ns	1

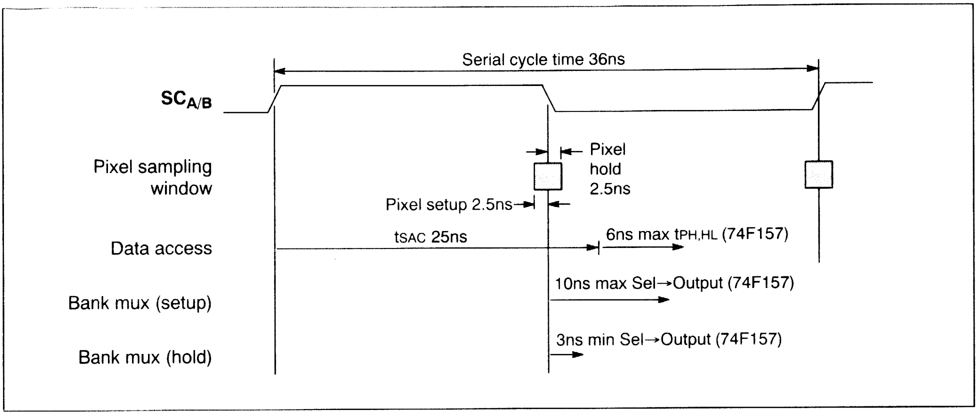
NOTE:
1 These timings apply to 2Mbit Micron VRAM devices

From Figure 6.4, it is apparent that, in addition to assessing the ability to meet the pixel setup time of 2.5ns (when the maximum buffer propagation delay and video RAM output turn-on time are considered), maintaining the 2.5ns pixel hold time is also important. The pixel hold time is determined by the minimum propagation delay of the 240/241 buffers added to the time it takes the video RAM serial outputs to turn on/off in response to $\overline{\textbf{SOE}}$ changing. Minimum delays for the buffers are quoted as 2ns and the minimum serial output turn-off time as 3ns. This therefore ensures that the pixel hold time will be met

For higher dot rate video systems the serial output turn-on/off times provided by currently available video RAMs are too long to allow the **$\overline{\text{SOE}}$** pins to be used for multiplexing between bank A and bank B. External multiplexors must therefore be used.

The most commonly required video frequency above 80MHz is 108MHz, to display a 1280×1024 screen at 60Hz refresh rate. This requires a serial port cycle time of 36ns for the true/full-color modes of the IMS G335, and thus 18ns for the time to multiplex between bank A and B. If the system shown in Figure 3.8 is implemented using 74F157s for the external multiplexors, then, using 25ns serial access video RAMs, the system can be realised to operate at the required 108MHz. Figure 3.11 shows a diagram using timings for the 74F157 and -8 video RAMs. Note that 30ns serial access video RAMs would be too slow for this system to meet the pixel setup time.

Figure 3.11. System timings for 108MHz interleaved implementation



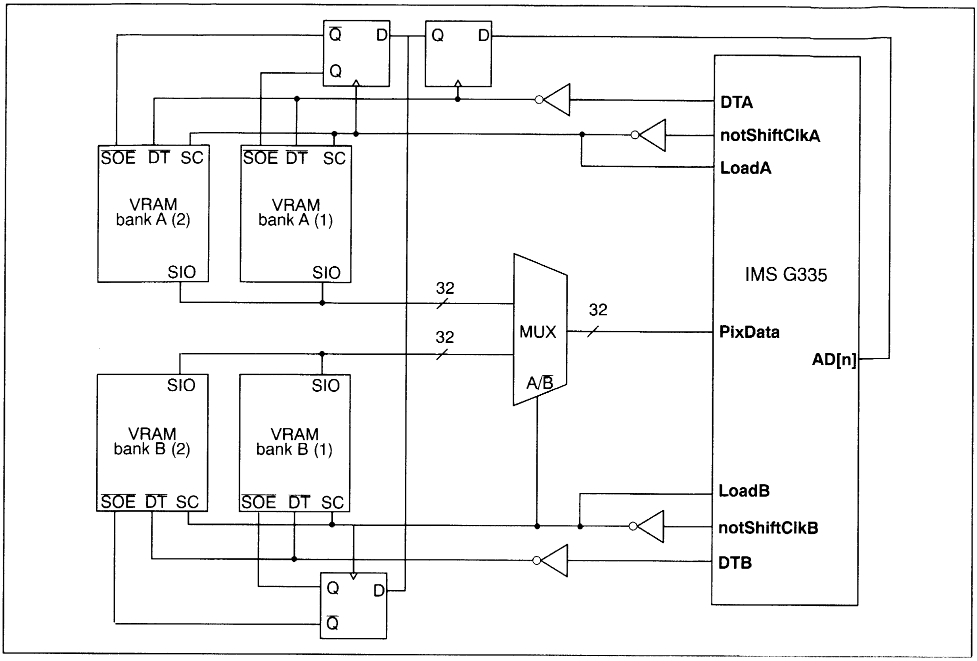
In order to run with 4:1 packed pixel multiplexing at 135MHz (the frequency required for a 75Hz refresh rate 1280×1024 display) the multiplexing system must implement fast SAM access video RAMs.

Imminent advances in video RAM technology will allow 4:1, 131MHz graphics systems to be designed since they will feature 20ns serial access times. Using such video RAM devices will allow the design methodologies applied in this section to fulfil the 135MHz requirements. New video RAM components also aid design definition of pixel setup time since serial access times are specified as less than the serial clock cycle time.

Interleaved mode multiple video RAM bank systems

Multiple sets of interleaved video RAM banks can be supported in a similar fashion to the system described for non-interleaved operation. This applies to both normal and split-SAM transfers, by extending the bank swapping hardware to provide independent control of the **$\overline{\text{SOE}}$**s for banks **A** and **B**, as shown in Figure 3.12. Whereas for non-interleaved operation, only **notShiftClkA** is used to clock the video RAM **SC** pins, the interleaving mechanism requires that bank A **SC** signals are clocked from **notShiftClkA**, and those of bank B from **notShiftClkB**. Hence the bank swap point needs to be controlled independently by **DTA**, **notShiftClkA** and **DTB**, **notShiftClkB** respectively.

Figure 3.12. Two sets of interleaved video RAM banks, using unsplit transfers



3.5 COMPATIBILITY ISSUES BETWEEN SAMPLING MECHANISMS

Compatibility between the IMS G332 and IMS G335 in both pixel sampling modes requires separate hardware support, implying some redundant logic between the implementations.

Consider the IMS G332 internal sampling implementation:

The first design issue with this mode is that video RAM serial access bandwidth is decreased from the optimal value due to the delay between the **notSerialClk** and the **notShiftClkA–B** signals. Ignoring skews between these signals for the moment, it is typical for the data to be shifted from the video RAM up to 8ns (typically 5ns) into a **notSerialClk** cycle, which in turn suggests that the pixel data sample referenced to **notSerialClk** occurs sooner than a **notShiftClkA** (or B) cycle time. But the placement of the sampling point with respect to **notSerialClk** attempts to limit the effect by delaying the pixel sample window with 1ns setup and 9ns hold around **notSerialClk**.

This delay actually provides slightly more SAM access time. The designer can verify the fact that in an interleaved framestore design using 74F241 inverter/buffers and 74F157 multiplexors, the video rate is limited to about 92MHz operation in 16bpp 2:1 multiplexing mode (a further 2:1 multiplexing is defined externally using the 74F157 multiplexors). Using 25ns access serial port video RAMs the minimum **notSerialClk** cycle time and thus maximum video rate can be calculated as follows:

	3.0ns	worse case skew between **notSerialClk** and **notShiftClkA–B**
+	8.0ns	maximum propagation delay through the inverter/buffer driving the SC clock inputs to the video RAM SAM port
+	25ns	to access the data from the SAM port
+	6.0ns	pixel data delay through the interleaving multiplexor
+	1.0ns	pixel data setup time
Total =	43ns	

With 4:1 interleaving (16bpp mode on the IMS G332) this defines a maximum pixel rate of 93MHz.

The hold time using this method is then calculated as follows:

	2.0ns	minimum propagation delay through inverter/buffer
+	8.0ns	maximum propagation delay through the inverter/buffer driving the multiplexor select input
+	2.5ns	minimum propagation delay through the multiplexor
–	3.0ns	worse case skew between **notSerialClk** and **notShiftClkA–B**
Total =	9.5ns	

IMS G335 internal sampling

The IMS G335 internal sampling mode is slightly improved over the IMS G332 sampling mode since it redefines the reference clock during sampling from **notSerialClk** to the relevant **notShiftClkA** (or B). Assuming the same architecture as above this allows the maximum frequency to be increased slightly to 100MHz, using 25ns serial access video RAMs, since serial to shift clock skews are eliminated. Note that the same setup and hold requirements are specified.

IMS G335 external sampling

Finally, the IMS G335 also has an external sampling mode which allows additional savings in propagation delays through various devices in the pixel sampling path. This extra saving is made by further re-defining the sample point to where the effect of external clock inverter/buffer logic (typically a 74F240 device) is eliminated. The maximum operating frequency is then derived as follows:

+	25ns	to access the data from the SAM port
+	6ns	pixel data delay through the interleaving multiplexor
+	2.5ns	pixel data setup time
Total =	33.5ns	

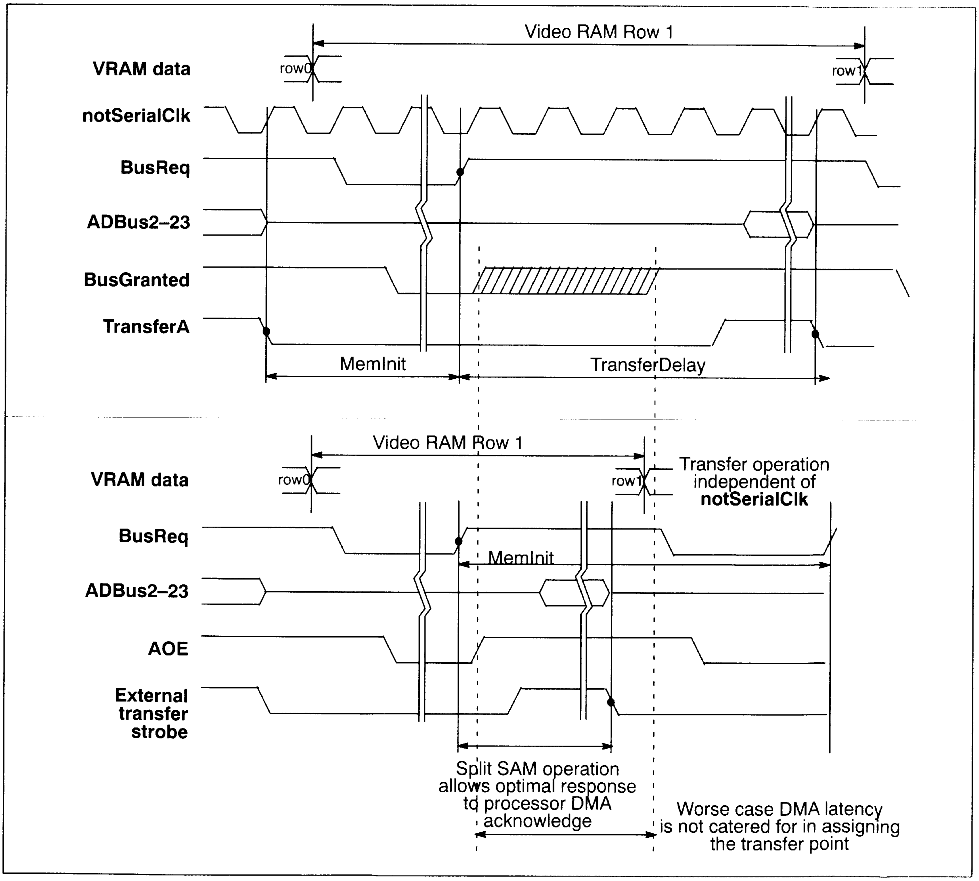
With 4:1 interleaving (16bpp mode on the IMS G332) this defines a maximum pixel rate of 118MHz.

The hold time is reduced substantially in the external sampling mode on the IMS G335 devices, allowing the multiplexor select line to be driven by a signal derived from **notShiftClkB**. Typically, the minimum propagation delays through the 74F240 and the select of the 74F157 provide adequate hold time.

3.6 SPLIT-SAM TRANSFERS WITH THE IMS G335

The split-SAM functionality provided by the IMS G335 provides a number of benefits. Systems which have a high maximum DMA latency duration will feature a highly improved system processor bandwidth over that possible with the IMS G332. With these devices the transfer point is controlled by theVideo Timing Generator (VTG) using the value TransferDelay. This transfer point must be resolved to within half a **notSerialClk** period to allow real-time read transfer operations supporting seamless midline updates. In order for the IMS G332 to perform successful transfer operations the maximum system DMA latency must be calculated in terms of screen units, even though the system processor may allow the DMA earlier. The scenario that exists for the IMS G332 transfer mechanism is that the point at which the DMA is granted is effectively ignored. If the processor allows a DMA almost immediately the **BusReq** signal from the IMS G332 is asserted, the DMA acknowledge to the transfer point (less the video RAM minimum random cycle time) is wasted time. This results in a loss of processor bandwidth.

Figure 3.13. Split-SAM allows optimal response to DMA acknowledgement



From the comparison of real-time and split transfers the duration between a DMA request being acknowledged and the exact point of transfer, omitting the minimum transfer cycle time (typically 160ns) consumes time that the processor could use for drawing.

A related problem with real time read transfer operations from the video RAM internal array to SAM is that of maintaining accurate placement of the transfer point. The transfer signal, generated by the CVC during DMA cycles, drives the $\overline{\textbf{DT}/\textbf{OE}}$ pin of the video RAM bank via combinational logic which allows the processor read strobes to support the $\overline{\textbf{OE}}$ function on random cycles to the video RAM. The combinational logic delay between the transfer strobes produced by the CVC and the signal applied to the video RAM's $\overline{\textbf{DT}/\textbf{OE}}$ pin becomes significant at **SC** cycle times of 30ns (131MHz video rate), since the placement of the transfer point is specified with parameters t_{TSL} (typically 5ns min) and t_{TSD} (typically 10ns min) with respect to the video RAM **SC** signal. These parameters, defining *transfer command to* **SC** *lead time* and the *first* **SC** *edge to transfer command delay time* respectively, result in a very small window, typically 10-15ns in which to place the transfer point.

In split-SAM mode the serial port of the video RAM can be controlled as two separate SAMs **A** and **B**. The principle of a swinging buffer is then applied, in which data is shifted out of SAM **A** whilst SAM **B** remains unrequired. When a SAM controller starts shifting serial data from SAM **B**, the transfer of new data into SAM **A** may occur whilst SAM **B** serial data is being consumed by the controller. The generation of the transfer strobes is defined by the designer. **DTA** and **DTB** are not used on split–SAM mode. This transfer mechanism is important in 135MHz video systems since, in split-SAM mode, the transfer point placement window translates into a duration equal to half a linetime, compared to 10-15ns (non-split real-time read transfers at 135MHz) in which to place the transfer point.

Video RAMs that support split transfers from their RAM to SAM can also be supported by a minor change in the bank swapping circuitry, as shown in Figure 3.14. Since the transfer can occur asynchronously to **SC**, we need to utilize the video RAM **QSF** output to tell us the earliest point at which to perform the bank swap. The circuit uses an enabled D-type flip-flop to swap the banks; the row transfer containing the change in address to select the next bank will always occur when **QSF** is high. Therefore as soon as **QSF** goes low, indicating the first **SC** clock edge for the new bank, the D-type is enabled and the following shift clock edge updates the bank select $\overline{\textbf{SOE}}$s ready to sample the first set of pixels.

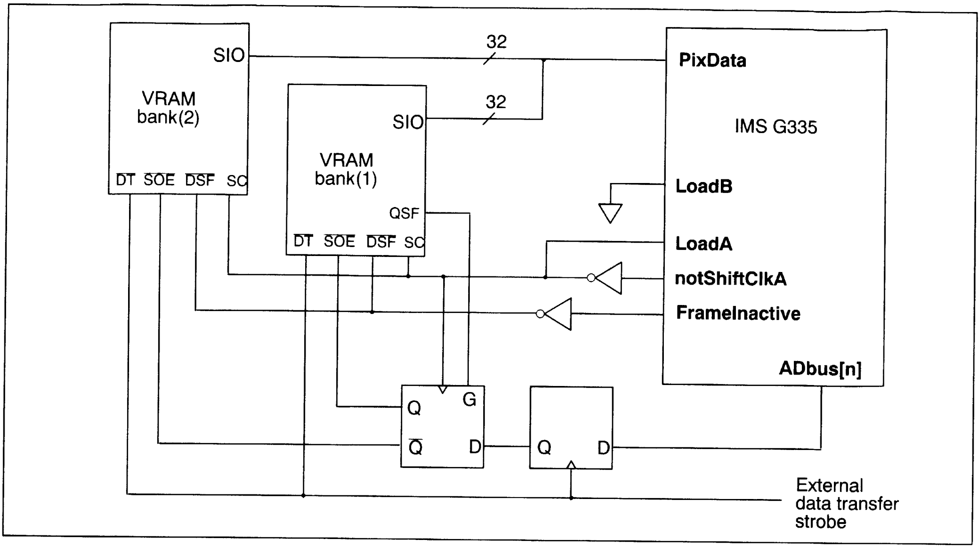
In such a system the video RAM **DSF** pin must be driven from an inverted **FrameInactive** from the G335, so that the first row transfer on each frame is an unsplit transfer, and all subsequent transfers are then split, as required for correct operation of the video RAM serial port. The use of the address increment output during transfer cycles is assigned such that transfers occur twice as often during the displayed screen as shown in Table 3.10.

Table 3.10. Control of split SAM video RAM increment

Register bit		Split SAM video RAM increment	
13	**12**	**First backporch in VBlank**	**During displayed screen**
0	0	not allowed	not allowed
0	1	256	128
1	0	512	256
1	1	1024	512

Since all the video RAM devices receive the same **SC** clocks, their **QSF** pins behave identically, and so only *one* needs to be connected to the enable input of the D-type.

Figure 3.14. Split SAM implementation with the IMS G335



A small disadvantage of using split-SAM transfers with the IMS G335 is the added complexity associated with the design of interleaved multiple bank swap hardware, which requires a defined **QSF** conditional input to enable the bank swap **notShiftClkA-B** clocks.

3.7 FRAMESTORE ARCHITECTURES

This section describes available framestore architectures in various Color Video Controller modes. It is important to understand the implementation of multiple banks of video memory which build the framestore:

Interleaved bank: This term describes a segment of the framestore video RAM corresponding to *two* fully populated serial ports multiplexed into a single pixel bus applied to the **PixData** pins of the CVC. Using 128K×8 video RAMs an IMS G332 or G335 system, for example, is configured to sample two 256×32-bit wide pixel data packets serially.

Multiple-set: A multiple number of interleaved banks. Thus in an IMS G335 design for example, requiring a display resolution of 1280×1024 with 16-bit pixels, a total of 2.5MBytes of framestore memory is required. This can be designed using 128K×8 devices in a three-set interleaved framestore, or using 256K×4 devices in a two-set interleaved framestore.

3.7.1 NON-INTERLEAVED MODE SINGLE VIDEO RAM BANK SYSTEMS

A variety of graphics systems can be designed using this architecture. Table 3.11 below gives a few examples of common screen formats and what can be realised without resorting to implementing further video RAM banks.

Table 3.11. Framestore utilization for a non-interleaved, single 32-bit wide video RAM bank

128K×8 video RAMs (0.5MByte)	800x600 @8bpp (1 screen)	640x480 @8bpp (1 screen)

256K×4 video RAMs (1MByte)	1024x768 @8bpp (1 screen)	800x600 @8bpp (2 screen)	640x480 @8bpp (3 screen)

The above table can easily be extended to the lower bits-per-pixel modes - 4, 2 and 1bpp; their memory requirements are simply 1/2, 1/4 and 1/8th of the memory needed for an 8bpp screen.

3.7.2 NON-INTERLEAVED MODE MULTIPLE VIDEO RAM BANK SYSTEMS

Two banks of video RAM extends the displayed screen format design possibilities beyond those described above in Table 3.11. Table 3.12 shows the options for the architecture shown in Figure 3.6.

Table 3.12. Framestore utilization for non-interleaved double video RAM bank

128K×8 video RAMs (1MByte)	1024x768 @8bpp (1 screen)	800x600 @8bpp (2 screen)	640x480 @8bpp (3 screen)

256K×4 video RAMs (2MByte)	1280x1024 @8bpp (1 screen)	1024x768 @8bpp (2 screens)	640x480 @24bpp (1 screen)

3.7.3 INTERLEAVED MODE SINGLE VIDEO RAM BANK SYSTEMS

The memory requirements for the 1280×1024 screen at 16bpp for the G335 exceed that provided by one set of interleaved video RAM banks; the next section describes how further sets of banks can be added to implement the required amount. Table 3.13 below shows which true color screen sizes are supported by one set of interleaved video RAM banks.

Table 3.13. True/full color framestore utilization for one interleaved video RAM bank

128K×8 video RAMs (1MByte)	1024x768 @8bpp (1 screen)	800x600 @16bpp (1 screen)	640x480 @16bpp (2 screen)

256K×4 video RAMs (2MByte)	1280x1024 @8bpp (1 screen)	1024x768 @16bpp (1 screens)	800x600 @24bpp (1 screen)

3.7.4 INTERLEAVED MODE MULTIPLE VIDEO RAM BANK SYSTEMS

In order to implement a 1280×1024 true color system for the IMS G335; a minimum two-set interleaved framestore is required if 256K×4 video RAM devices are implemented as shown below in Table 3.14.

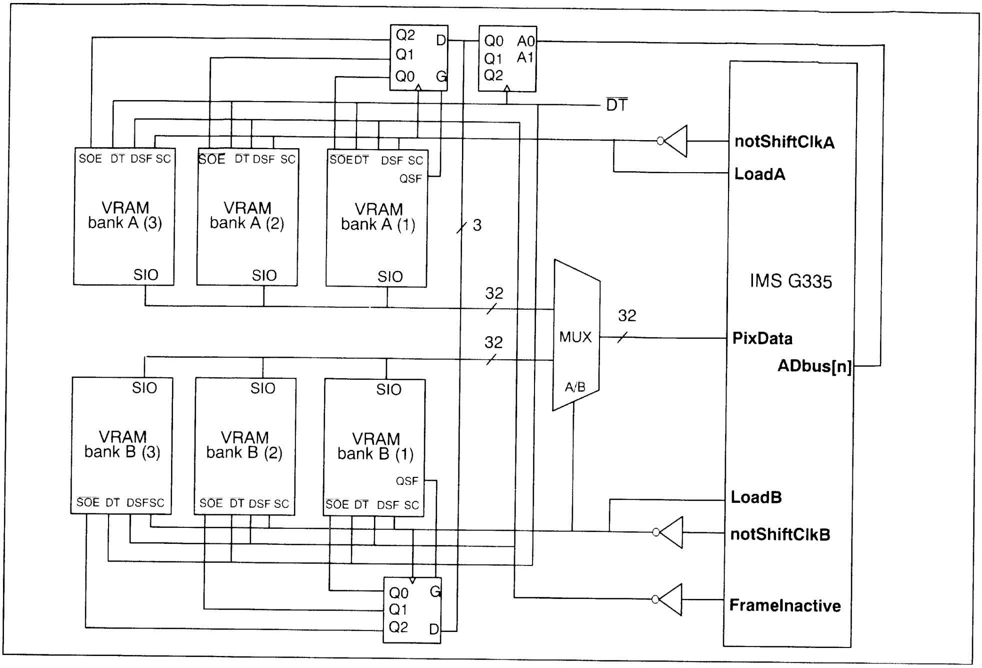
Table 3.14. True/full color framestore utilization for two interleaved video RAM banks

128K×8 video RAMs (2MByte)	1280x1024 @8bpp (1 screen)	1024x768 @16bpp (1 screen)	800x600 @24bpp (1 screen)

256K×4 video RAMs (4MByte)	1280x1024 @16bpp (1 screen)	1024x768 @16bpp (2 screens)	800x600 @24bpp (2 screen)

At 1280×1024 screen resolution with 16-bit pixels, 2.5MBytes of the framestore is unused, although at 1024×768 a second screen buffer is available. In graphics systems not requiring display buffers however, system economics would dictate a lower density framestore. One criteria that the CVC true color modes specify is that an interleaved video RAM architecture is completed external to the device such that half of the serial port cycle time is pipelined. This interleaving structure dictates that the number of banks in total is even. Thus in an IMS G332/5 design using 128K×8 video RAMs the serial ports from two 0.5MByte banks are interleaved onto the **PixData** bus; any further sets of interleaved banks will result in an additional number of 1MByte video RAM arrays. The minimum required to implement 1280×1024 screen resolution with 16-bit pixels thus requires a three-set interleaved video RAM configuration as shown in Figure 3.15.

Figure 3.15. Three sets of interleaved video RAM banks, using split transfers



3.8 OTHER ENHANCEMENTS AND COMPATIBILITY ISSUES

3.8.1 CONTROL REGISTER B

The new functions of the IMS G335 described in this Application Note are selectable through Control Register B. This register was defined in the memory space of the IMS G332, at address #X070, and required #0000 to be written in the startup sequence. This register in the IMS G335 must still be written to #0000 as part of the startup sequence before writing control data. The additional modes, over those available with the IMS G332, are selected through values written to Control Register B:

- External sampling
- Sync on green DAC only
- HSync during flyback
- Split-SAM operation
- DAC comparator selectable for red, green and blue
- Turn on comparators
- Checksum on zero palette
- Sync pins tristate
- **notVSync** active high
- **notCorHSync** active high
- Complement (XGA compatible) cursor
- XGA cursor selection
- Even field interlace

3.8.2 TRUE COLOR MODES (15BPP AND 16BPP)

The IMS G335 does not support the 6:6:4 RGB pixel format. The replacement format is the XGA compatible 5:6:5 format. Each pair of pixel ports (A and B, and C and D) supplies a two-byte pixel value which is split into red, green and blue fields. The ports are used in the order A,B then C,D. In 15 bits per pixel mode bits 0-4 are blue, 5-9 are green, and 10-14 are red (RGB format 5:5:5). In 16 bits per pixel mode bits 0-4 are blue, 5-10 are green, and 11-15 are red (RGB format 5:6:5).

3.8.3 SCREEN UNIT DEFINITION WITH THE IMS G335

The IMS G335 includes 24bpp format, providing support for 1, 2, 4, 8, 15, 16 and 24bpp. Since the IMS G335 pixel bus is 32 bits wide no internal multiplexing of 24bpp is possible. However, in interleaved mode, 24-bit pixel rates of twice the current maximum video RAM clocking speed can be attained. Up to approximately 80MHz can be achieved (using a hardware multiplexer external to the device) using this method, giving support for 24-bit full color, 1024x768 resolution at 72Hz screen refresh rates. Up to around 40MHz can be achieved in a non-interleaved system, giving support for TV broadcast standards (NTSC, PAL, etc.) at 24 bpp from a single bank of memory. If the video rate is low enough, the video RAM serial output enables can be used to supply pixel data from alternate memory banks without the use of external hardware.

Note that in 24 bpp non-interleaved mode:

1 Screen Unit = 1 pixel

and in 24 bpp interleaved mode:

1 Screen Unit = 2 pixels

When programming the screen description line timing parameters, it must be noted that 1 Screen Unit is no longer 4 pixels long as in all other bits per pixel modes and as standard on the IMS G332. The parameter duration must therefore be calculated using the values given above.

3.8.4 CURSOR PIXEL OPTIONS

Three options are available with the IMS G335 controller to determine cursor pixel appearance, depending on the setting of Control Register B, bits 12 and 13, as shown in Table 3.15.

Table 3.15. Selection of cursor pixel options

Control Reg. B		Function
Bit 13	Bit 12	
0	0	Ordinary cursor, 3 colors + transparency (IMS G332/364 format)
0	1	2 colors + transparency + complement (pixel complement option)
1	0	Invalid
1	1	XGA cursor format

The first option is the standard format implemented in the IMS G332 and is the default option.

The pixel complement option replaces one of the cursor colors with 'pixel complement'. This generates the complement (all bits inverted) of the pseudo or true color pixel already on the screen, ensuring that cursor pixels programmed with this will always be visible whatever the background. This option is selectable through bit 12 of Control Register B.

The XGA cursor option maps the cursor palette according to the XGA standard. The XGA cursor maps the palette in a different way to that of the IMS G332 and includes pixel complement. This option is selectable through bit 13 of Control Register B.

3.9 CONCLUSION

The design of the IMS G335 Color Video Controller carefully caters for graphics systems featuring the IMS G332 devices. The IMS G335 is functionally and pin compatible with the IMS G332 allowing system features to be upgraded for external sampling, improved micro port performance, split-SAM operation and other enhancements including support for XGA compatible cursor and pixel formats, software selectable pixel bus width, additional synchronization, DAC comparators and a checksum test mode

Two new pins are provided on the IMS G335 which were previously **HoldToGnd** inputs on the IMS G332. The support of these inputs is all that is required to implement the IMS G335 controllers new external pixel sampling mechanism. The **LoadA-B** inputs are driven from the **SC** inputs to the video RAMs and can thus be designed into IMS G332 based board products by simply routing two new tracks. The implementation of the IMS G335 controllers enhance system performance by providing split-SAM support, improved DMA scheduling and faster read/write access through the micro port, as well as a 64×64×2-bit hardware cursor on-chip.

This document illustrates how all design enhancements of the IMS G335 can be accommodated with forethought and simple modification of existing support logic. This note also illustrates that redundant logic with the enhanced modes of the IMS G335 can be minimized since it employs segments of support logic that would have been implemented in IMS G332 designs.

As part of this conclusion, a number of points are re-iterated as follows:

- Be careful defining the address increments in non-interleaved and interleaved framestores. Although interlaced display and split-SAM increments may appear to confuse things, the initial increment selected for the framestore design will not change because it is selected on the basis of the framestore architecture.

- The pixel sampling element of the design is critical in multiple-set bank systems. The status of the select line, derived from **notShiftClkB** is designed to perform this function, although the actual delay between an IMS G332 system and an IMS G335 with internal sampling varies with the definition of pixel setup and hold times.
- Design primarily to support the external sampling mode. To do this simply route the **LoadA** and **LoadB** tracks on the board layout. This allows simple upgrade to the higher performance pixel sampling mechanism with a slot-in of the G335 part and a new value written to Control Register B to enable external sampling.
- Split-SAM support can be obtained with little overhead in external logic. The main factor in achieving an upgrade to this mode is the use of enabled D-type latches and a method for generating the transfer strobe. The latter can be a small addition to the state machine logic generating the RAS and CAS signals with the **DTA/B** signals being held low by the IMS G335 in split-SAM mode.

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

MIGRATING G300x DESIGNS TO A G335 SOLUTION

GRAPHICS APPLICATIONS GROUP, BRISTOL

4.1 MIGRATING G300x DESIGNS TO A G335 SOLUTION

4.1.1 INTRODUCTION

In 1989 INMOS (now part of SGS-THOMSON) launched the world's first Color Video Controller (CVC), the IMS G300. This integrated a Video Timing Generator (VTG), pixel multiplexor and Phase Locked Loop (PLL), which simplified graphics system design and allowed flexible pixel resolutions. Three versions of the IMS G300 were released, the G300A which only performed 8 bits per pixel (bpp) and 24bpp, G300B which allowed 1/2/4/8 and 24bpp and G300C which had improved current reference inductive decoupling.

SGS-THOMSON has now announced the IMS G335 CVC which integrates more functions, allows greater pixel frequencies than the original G300 and offers greater system flexibilities.

This document details the hardware and software changes that are applicable when migrating G300 graphics systems into G335 based graphics designs.

4.1.2 MINIMAL REQUIREMENTS TO MIGRATE A G300 DESIGN TO A G335

When migrating G300 designs to a G335 solution the following areas should be studied:

- **Micro Port**: the G335 has a slightly different, improved micro port to that of the G300, which is described in Section 3.1.
- **Pixel Port**: for simple replacement of the G300, changing the framestore architecture to a G335 solution should prove to be a trivial exercise. See Section 3.4 for more detail.
- **Software:** the G335 provides extra functions and a changed register address map to that of the G300, These are described in Section 5.
- **SAM transfers**: the G335 is completely compatible with G300 with respect to SAM transfers, but also incorporates support for split SAM transfer, which is described in Section 6.2.

4.2 DIFFERENCES BETWEEN IMS G335 AND IMS G300

4.2.1 HARDWARE

The IMS G335 is capable of running at pixel frequencies of up to 135MHz compared with 110MHz as supported by the G300C. The G300 is packaged in a 100 pin ceramic quad flatpack. The G335 is packaged in a 100 pin ceramic quad flatpack, but is not pin compatible with the G300.

Block diagrams of the G300 and G335 are shown in Figures 4.1 and 4.2 respectively.

Figure 4.1. IMS G300 Block Diagram

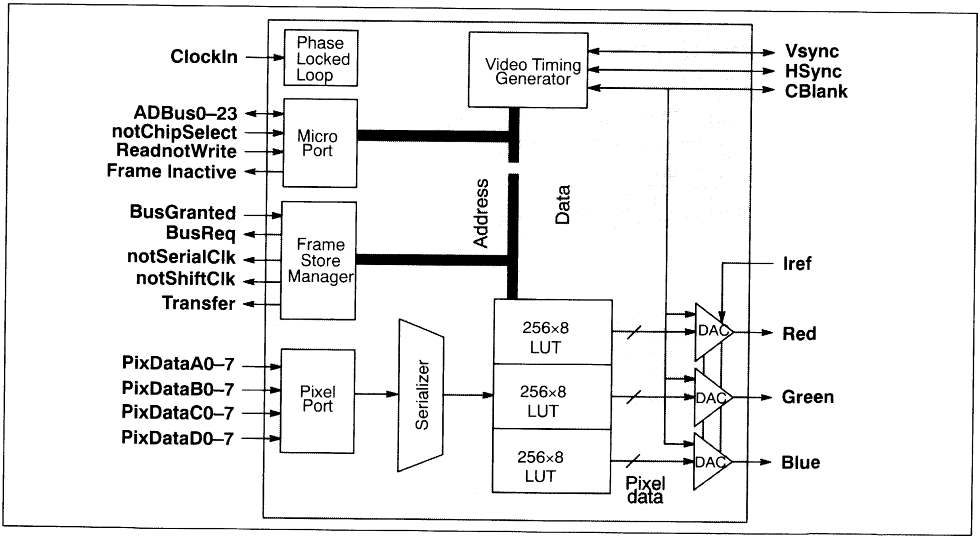
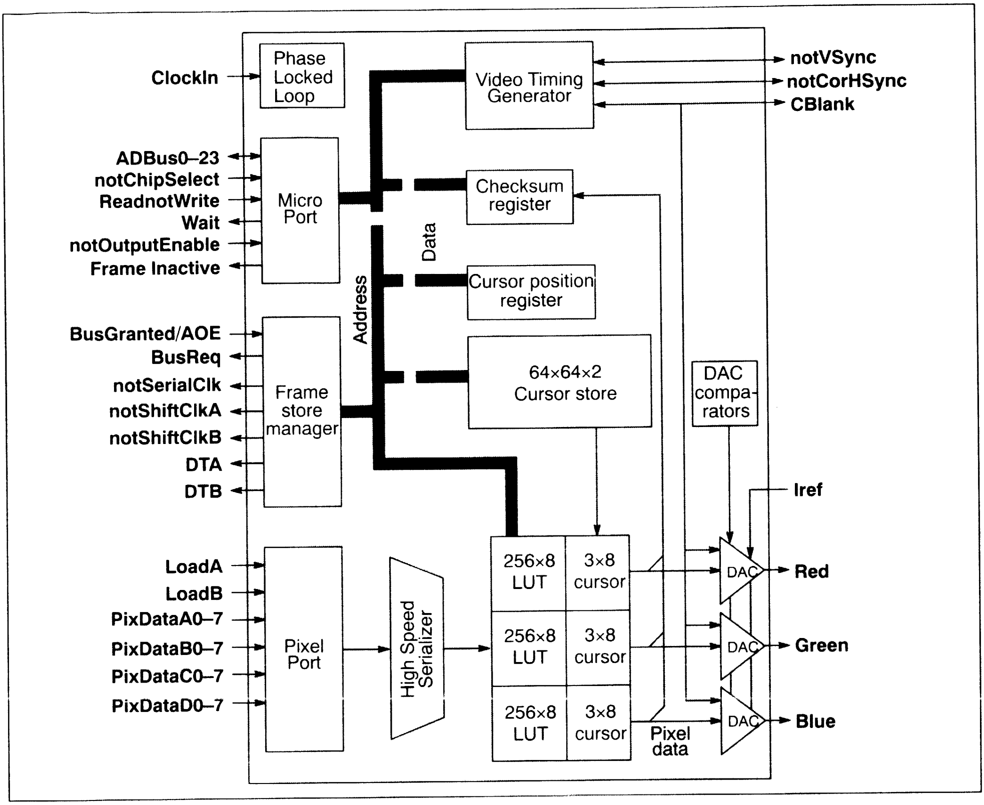


Figure 4.2. IMS G335 Block Diagram



4.2.2 PIN DIFFERENCES BETWEEN G335 AND G300

The G335 is very similar in pinout structure to the G300 but with the following extra pins:

Wait:

Allows microprocessor initiated read or write cycles to be synchronized into the G335 extended times of read or write accesses to device. The G300 processor interface required logic to be incorporated to allow correct access to the G300 for all pixel frequencies. The introduction of the **Wait** pin on G335 simplifies the design of the processor interface for all frequencies.

notShiftClkA/B:

These clocks run in anti-phase of each other. Two shift clocks are present, as opposed to one shift clock on the G300, to allow VRAM bank interleaving. If VRAM bank interleaving is not required, as in a G300 design, then **notShiftClkA** would be used as the standard SAM (Serial Access Memory) shift clock. Note that these clock signals should be buffered with an inverting buffer.

DTA/B:

Data transfer strobes. These strobes replace the **Transfer** strobe that was originally on the G300, but their functions are similar. They control the exact moment at which a transfer of data from the DRAM to the SAM occurs. Two strobes are required if VRAM interleaving is used. These strobes are synchronized to the shift clock. For a non-interleaved VRAM bank graphics system, **DTA** is used as a replacement for the **Transfer** strobe.

LoadA/B:

These strobe inputs have no equivalent function on the G300. In normal operation, the pixel port is sampled on an internally generated signal that is derived from **notShiftClkA**. In operations similar to G300 designs, the **LoadA/B** inputs can be tied low.

For external sampling, pixels are sampled according to the strobes on **LoadA** and **LoadB** input pins. This mode allows higher sampling speeds to be achieved through shorter VRAM data set up and hold times. For all new designs it is recommended that external sampling is implemented.

When using external sampling, the pixel data is sampled on the rising edge of **LoadA/B**. If an interleaved framestore is not required then **LoadB** should be held to ground.

CBlank:

This pin, present on G300B and G300C but not on the G300A, can be set as either an input or output. If set as an output, it can be used to control externally generated VRAM shift clocks by ANDing this signal with the clocks, thereby generating a clock disable signal. It could also be used in this mode to supply blanking information to an external Palette-DAC.

If this pin is used as an input then an external blanking source must be provided. This can either be internally ORed with the internally generated Blank source or used without modification.

notOutputEnable:

This pin is not present on the G300x. This signal turns on the output buffers during a read cycle. It must be pulsed in order to terminate the access. Read cycles can be extended indefinitely by either holding off the pulse or holding this signal low.

4.2.3 ARCHITECTURAL DIFFERENCES BETWEEN G300 AND G335

The core of the VTG remains very similar to the G300. The main difference between the G335 VTG and the G300 is that the G335 devices support VRAM bank interleaving, which supplies two shift clocks and inputs to control latching of pixel data externally. The VTG on the G335 now supports split SAMs, which is not a feature on the G300.

The G335 allows pixels to be defined as 1,2,4,8,15,16 and 24bpp. The original G300A only allowed 8 and 24bpp, but G300B and G300C allow 1,2,4,8 and 24bpp. For 24bpp the G300B/C had to be put into a non pixel serializing mode (mode 2) which required an external clock source to provide a shift clock frequency, i.e. the internal PLL would not be used. On the G335 an external shift rate frequency is no longer required to produce 24bpp as the internal PLL can be used.

When in 24bpp (or 15,16bpp) mode, colors can be gamma corrected by programming the appropriate values in the lookup tables on G335. This could not be done on G300A as 24-bit pixels were passed directly to the DACs.

The G335 provides a 64×64 by 2-bit cursor, with transparency and pixel complement options. A 3 × 24-bit palette is available to define cursor colors. Cursor pixel definition can either be standard (3 color + transparency), 2 colors + transparency + complement (XGA cursor format). No external signals are required or generated for cursor operation.

On the G335 a checksum register is present, which allows the system manufacturer to verify the board and correct operation of the part by reading the G335 calculated checksum value and comparing with the theoretical calculated value.

4.3 DESIGN CHANGES REQUIRED FOR A G335 IN A G300 GRAPHICS SYSTEM

4.3.1 MICRO PORT

The following pins are common to both the G335 and G300 micro ports:

- **BusReq**
- **BusGranted/AOE**
- **ReadnotWrite**
- **notChipSelect**
- **ADBus0–23**

Extra pins on the G335 micro port are:

- **notOutputEnable**
- **Wait**

4.3.2 MICRO PORT ACCESSES

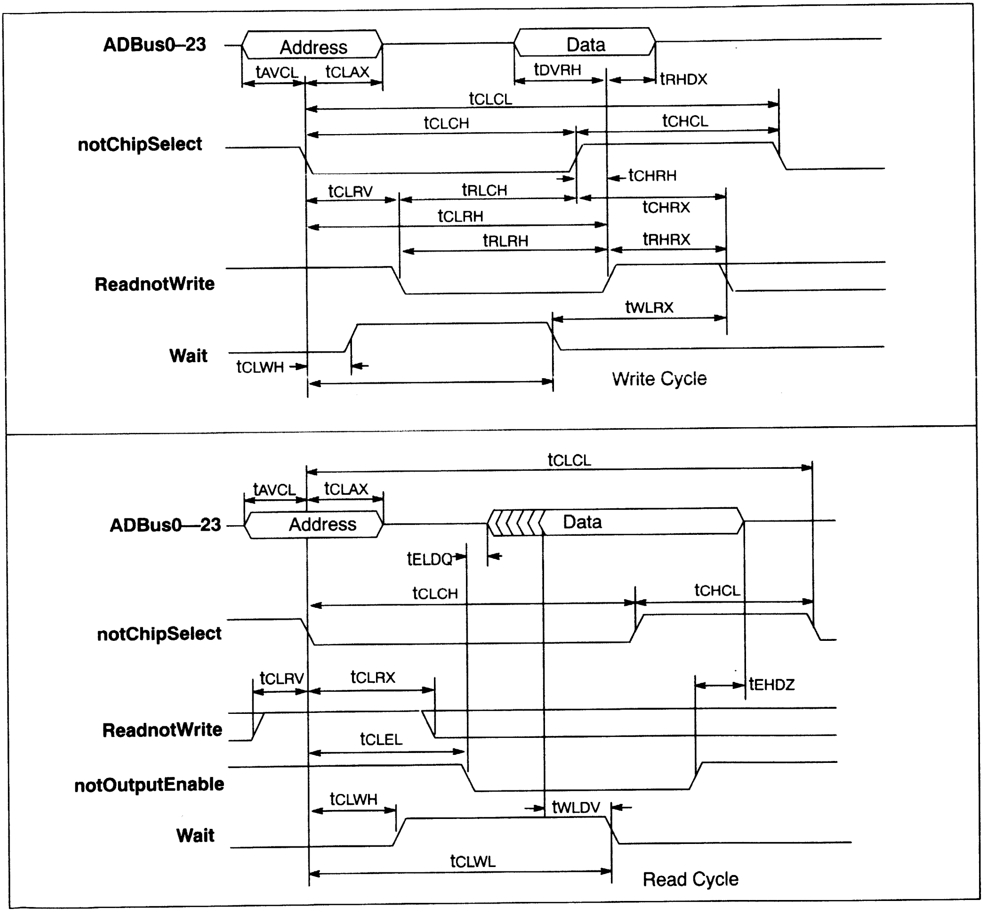
When writing to the G300 the minimum low time of **ReadnotWrite** is 4 periods of serial clock and the minimum time between successive write accesses is 1 serial clock period + 20ns. For reading, active low time of **notChipSelect** should be not less than 3.5 serial clock periods + 20 ns. Therefore the access time and cycle time are dependent on the pixel frequency.

The time between successive access times can be quite easily controlled in software, but to meet the criteria that active low times of **ReadnotWrite** and **notChipSelect** should not violate stated timings, the processor access cycles are required to be altered such that it is within specification. Certain designs have programmed the processor memory interface parameters so that processor memory accesses do not violate the timings for accesses to the G300. Unfortunately this is fixed for a certain pixel frequency, and if lower resolutions (with lower pixel frequencies) are required, then the memory interface needs to be reprogrammed.

It is possible to build a dynamic wait state generator that counts serial clock cycles and provides a relevant wait signal back to the processor on every G300 access. This allows the processor to access the G300 at any pixel frequency and meet the timings specified.

On the G335 an external dynamic wait state generator is not required. A **Wait** signal is provided that allows the processor to meet the timing criteria of the G335. A typical read and write access to the G335 is shown in Figure 4.3. See [3] for definition of timing parameters.

Figure 4.3. Timing diagram of read and write accesses to G335



For a write access, if the **Wait** signal is fed back to the processor to synchronize the access[2], the minimum active low time (t_{RLRH}) of **ReadnotWrite** can be met.

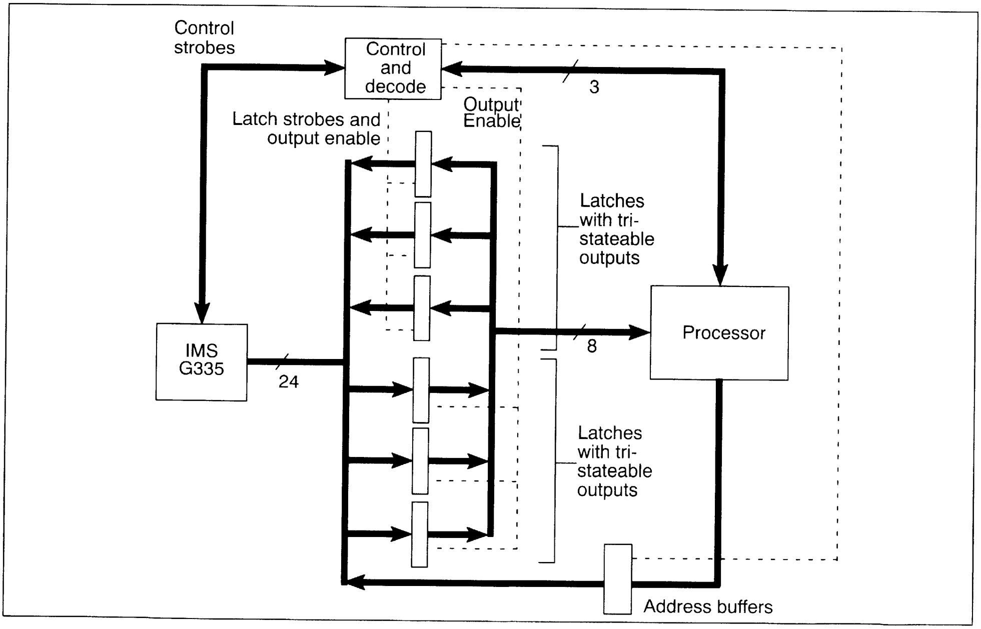
For a read access, the Wait signal synchronizes the access to the serial clock frequency and meets the required timings. **notOutputEnable** must be pulsed in order to terminate the access, this signal, when active low, turns on the output buffers of the G335. Read cycles can be extended indefinitely by holding off the pulse on **notOutputEnable** or holding this signal low. **notOutputEnable** could be a decoded read strobe from the microprocessor.

[1]. The **Wait** signal may have to be synchronized into the processor clock and may require external latching by the processor clock, but most processors have asynchronous memory interface logic which do not require external latching. It is suggested that the user checks with the appropriate processor manual to ensure compatibility.

4.3.3 8-BIT BYTE MODE ACCESS

The G300x provided a means of accessing the microprocessor interface via an 8-bit wide data path. Instead of one 24-bit wide data access, three 8-bit data accesses are performed, with data being read or written from or to **ADBus0–7** of the G300. The G335 devices do not provide 8-bit byte access mode. External latches and control circuitry would have to be modified from their present use in a non-multiplexed address data bus to incorporate an 8-bit byte mode access on the G335. A suggested block diagram for implementing 8-bit accesses to the G335 is shown in Figure 4.4. Note for simplicity, no address comparison is performed (ie to check if the microprocessor is accessing a different address location within a 3 byte access sequence).

Figure 4.4. Block diagram for implementing byte mode access on IMS G335


Control strobes
Control and decode
3
Latch strobes and output enable
Output Enable
Latches with tri-stateable outputs
Processor
IMS G335
24
8
Latches with tri-stateable outputs
Address buffers

4.3.4 PIXEL PORT

The G335 provides the same number of pixel port pins as the G300A/B/C.

Generally G300 graphics designs do not provide VRAM interleaving. However, the G335 does support VRAM interleaving and if required can assist in an easier implementation.

For a graphics system using 1Mbyte of VRAM without split SAM support, migrating a G300 to a G335 design is very simple. As VRAM bank interleaving is not used, **DTA** on G335 is used as master transfer strobe and **notShiftClkA** as master shift clock. **notShiftClockB** and **DTB** are ignored. **LoadB** is tied to ground. A block diagram of the pixel port for non-interleaved VRAM is shown in Figure 4.5. Note that **LoadA** is the inverted **notShiftClockA.**

If 2Mbytes of VRAM are required in the graphics system, then interleaving mode on the G335 could be used as shown in Figure 4.6. If external multiplexing of VRAMs is performed by using the serial output enable function (SOE) found on most VRAMs, then a careful study of output enable turn on and off times should be done to calculate maximum pixel frequency. For more information on interleaving of VRAM banks consult [1] .

Figure 4.5. Simple 1Mbyte graphics system using IMS G335

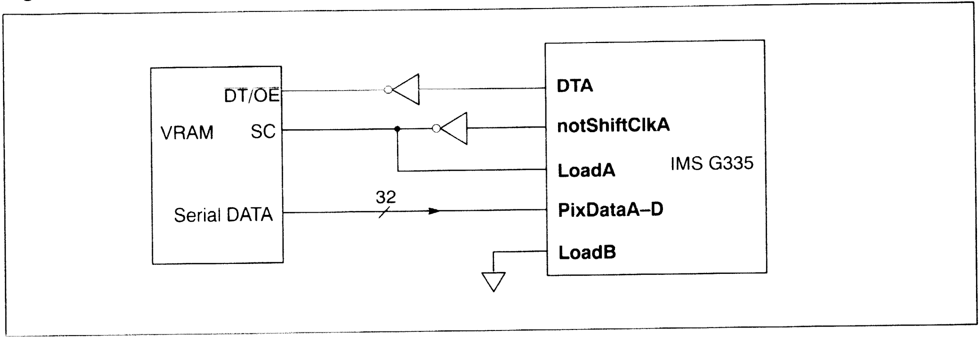
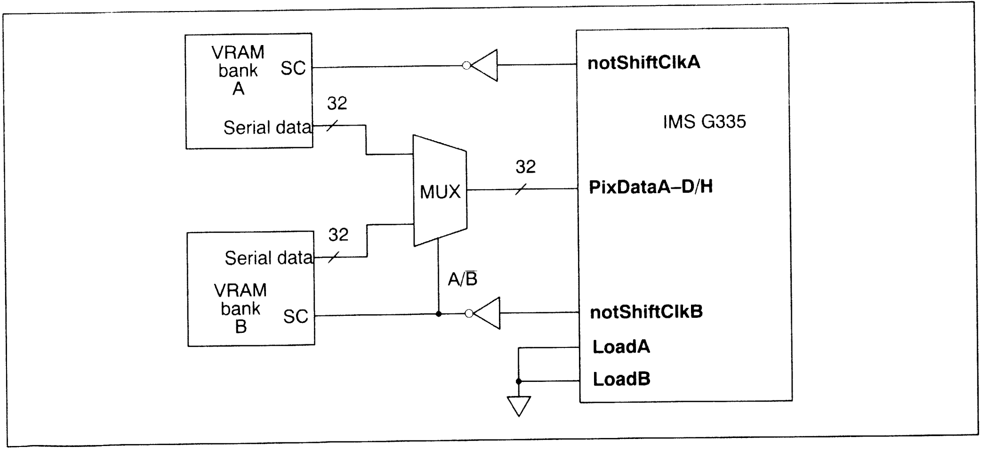


Figure 4.6. Interleaved VRAM



4.4 VRAM CONTROL

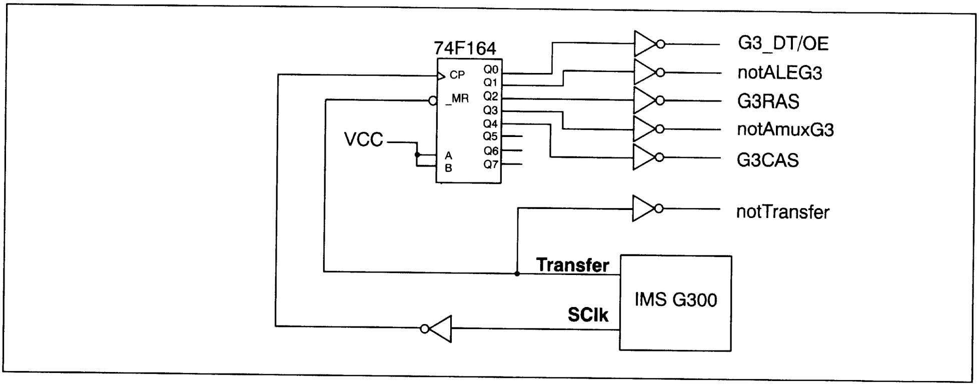
4.4.1 TRANSFER CYCLE

The SAM transfer cycle on the G335 is very similar to that found on the G300. Two transfer cycles exist on the G335, split SAM and normal SAM transfer cycle.

4.4.2 NON SPLIT SAM TRANSFER CYCLE

The G335 provides several strobes that signal when a serial access memory transfer is required. The strobe generation presented to the VRAMs to control a SAM update is left to the user to implement, but the simplest method (used on G300 designs) is to use a shift register that counts serial clock cycles as shown in Figure 4.7. This circuit provides RAS, CAS, multiplexing control strobes and DT/OE strobes to generate a SAM update cycle. Unfortunately, locking the strobe generation into the serial clock means that the length of time to do a SAM update is dependent on the pixel frequency, and for slow pixel rates, this time could be quite considerable. Also, the G3xx holds the VRAM address bus for a fixed number of **notSerialClock** cycles, defined by the contents of the TransferDelay register. The TransferDelay value is also dependent on the maximum DMA latency of other bus masters in the system, this amount of time can be quite sizeable for low pixel frequency displays (such as PAL or NTSC).

Figure 4.7. SAM update circuit, for non-split SAM transfers



4.5 SOFTWARE ISSUES

Programming the G335 is very similar to programming the G300. The main differences are that there are more functions available, two new registers have been added and certain registers have moved location.

4.5.1 RESET

The reset procedure for G335 is the same procedure as reset for G300. Reset causes default values to be set in the Boot Location and Control Registers

4.5.2 BOOT LOCATION

The Boot Location present on the G335 is slightly different to that found on the G300, the differences being:

- Boot Location has changed from address #X1A0 on the G300 to #X000 on the G335
- The G335 provides an option for setting a micro port address alignment of 32 bits or 64 bits. This is not present on the G300.

The change of Boot Location address on the G335 to #X000 means that the color palette, present from address #X000 to #X0FF on the G300, now resides at address #X100 to #X1FF. Reading and writing to the palette are still performed in the same way as on the G300.

4.5.3 8-BIT BYTE MODE MICRO PORT ACCESS

The G335 does not provide a byte mode access so the relevant field is not required in any control registers. If a byte mode access has been implemented in external hardware and no address checking is performed, then the software should be checked to make sure that it is compatible with the new interface scheme (especially in multitasking or interrupt driven systems).

4.5.4 CONTROL REGISTERS

The G300 provided a Control Register at address #X160 which allowed configuration of the device, such as DMA enabled, bits/pixel etc. The G335 has two control registers, called Control Register A and Control Register B. These reside at addresses #X060 and #X070 respectively. The G335 Control Register A is not too different from the G300 Control Register, but Control Register B defines the usage of the extra G335 features. Control Register B can be programmed such that the G335 behaves in a similar function to the G300. The composition of these registers can be found in [2].

4.5.5 PROGRAMMING THE VTG

Programming the VTG on the G335 follows the same procedure as for the G300; the VTG must be disabled from running by a setting bit 0 in Control Register A to 0, the VTG parameters are then written and the VTG allowed to run by setting Control Register A bit 0 to one.

Some extra registers are provided on the G335 that were not present on the G300. These are VPreEqualize, and VPostEqualize. VPreEqualize and VPostEqualize are user programmable on the G335; on the G300 these timings were set by VSync parameter.

4.6 EXPLOITING EXTRA FUNCTIONALITY ON THE G335

4.6.1 ENHANCED AND EXTRA FEATURES

The G335 supports the following enhanced and extra features, some of which are described later:

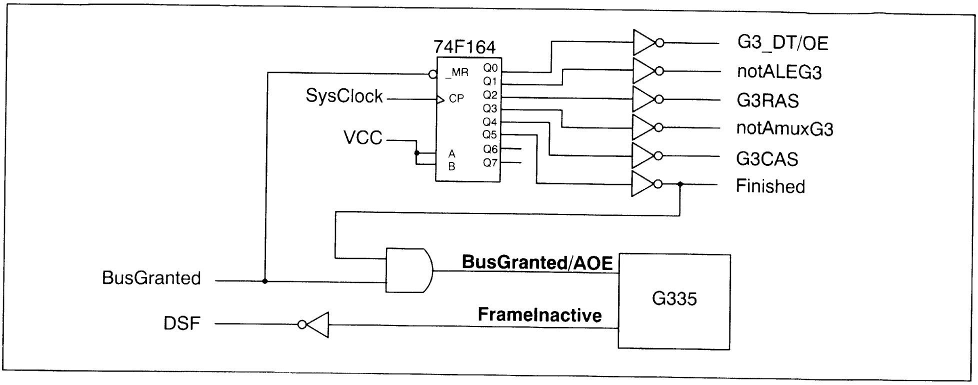
- Programmable polarity of HSync and VSync strobe generation
- Continuous HSync during frame flyback, both in Master and Slave mode
- Tri-stateable sync pins
- 1,2,4,8,15,16 and 24 bits per pixel
- 32 or 64 micro port address bit alignment
- Cursor, 64×64× 3 color cursor
- Split SAM support
- VPreEqualize and VPostEqualize
- Checksum register
- Interleaving support
- Pixel port external sampling
- DAC output comparators
- Improved DMA system by reducing bus mastering times

4.6.2 SPLIT SAM TRANSFER CYCLE

The G335 supports split SAM update cycles to increase the system performance by increasing valuable processor bandwidth to the VRAMs. The use of split SAM allows simplified system design and reduces bus mastering time. Split SAM VRAMs have their serial shift register split into two, one half being updated while the other half is shifting out. The transfer window now becomes less critical than in a non-split SAM system, as a transfer to one half of the SAM can occur when data is being read from the other half of the SAM (for more information on split SAM transfer consult relevant VRAM datasheet). Also the generation of the RAS and CAS strobes that are presented to the VRAMs could be generated from any arbitrary clock signal, such as the processor clock or generic system clock and hence is not synchronous to the pixel frequency of the graphics display.

On the beginning of every field, a non-split SAM transfer is performed as defined for proper operation of the VRAM, then during successive SAM updates, the split SAM transfer mode is used (if split SAM mode is enabled on the CVC). Figure 4.8 shows a block diagram for implementing split SAM update. The DSF pin is used to signify a Split SAM transfer; when low, normal SAM transfers occur, when high, a split SAM transfer is performed. **FrameInactive** can be used to instruct the VRAMs whether to do a split mode transfer or a normal transfer. An in depth discussion of using a G335 and split SAM transfers can be found in [1].

Figure 4.8. Split SAM update circuit



4.6.3 24 BITS PER PIXEL MODE

The G335 allows an easier implementation of 24 bits per pixel true color mode than previous CVCs. Unlike the G300, which required a mode 2 operation to implement true color, the G335 requires no such mode. True color mode can be implemented using a generated clock from the on-chip PLL similar to normal modes, with all VRAM management being performed as normal. Hence a graphics system could be designed around a pseudo color system but allowing true color operation without any hardware modification, just by programming appropriate registers on the G335.

4.6.4 VRAM BANK INTERLEAVING

VRAM bank interleaving, previously mentioned, allows twice the amount of VRAM to be installed through a common pixel port. For an IMS G335 system 4Mbytes of VRAM could be used by VRAM bank interleaving.

VRAM bank interleaving could effectively be implemented by using the SOE (Serial Output Enable) signal commonly found on VRAMs, whereby each bank's serial outputs are enabled or disabled in anti-phase of each other. Using this method requires no external multiplexing hardware, but close attention to the turn-on and turn-off times of the VRAMs used should be made to determine the maximum pixel frequency that can be obtained in the graphics system.

4.6.5 CURSOR

The G335 has an on-chip 64×64×2 hardware cursor, which requires no external memory or special controls and minimal processor requirements. All that is required to operate the cursor is to program the cursor mode operation, load a cursor pattern on the G335 and update cursor coordinates accordingly. Cursor colors are independent of the main palette and have to be defined using the registers provided.

4.7 CONCLUSION

The IMS G335 CVC is not too dissimilar to the original IMS G300. To convert designs based on the G300 to the G335 requires only a small number of modifications if the extra features present on the G335 are not required. However, even if the extra features are required (i.e. VRAM bank interleaving, split SAM support) then the differences in design are not considerable and can be easily achieved.

References

1 *32-bit Color Video Controller upgrade compatibility: a design guide*
Applications Note (Chapter 3 in this volume)
Graphics Applications Group
Document number: 72 TCH 129 00

2 *IMS G335 Datasheet*,
Computer graphics databook, 3rd Edn.
SGS-THOMSON,
Document number: 72 TRN 204 02

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

HARDWARE DESIGN WITH CVCS

5.1 INTRODUCTION

To obtain the best possible picture quality from SGS-THOMSON Color Video Controllers (CVCs), it is essential that they operate in a quiet electrical environment and that their input and output signals comply strictly with the requirements specified in the relevant datasheet. This document is intended to help the graphics system designer meet those requirements and hence realise the full potential for picture quality inherent in the CVC.

5.2 POWER SUPPLY RECOMMENDATIONS

It is essential that the device is provided with an adequately decoupled power supply. The greatest care with the other signals will be to no avail if the power supply is noisy.

First and foremost, a PCB with ground and VDD planes should be used. This minimizes the inductance and resistance in series with the power source, resulting in less noise than that which would result if the power rails were routed on the signal layers. An added bonus is that that the power planes act as a shield between signal layers, reducing crosstalk. The benefits of this will be dealt with later.

A 47µF tantalum capacitor should be connected between **GND** and **VDD** to decouple low-frequency supply noise. This should be placed as close to the device as possible. In addition, 0.1µF ceramic capacitors should be placed close to each pair of **GND** and **VDD** pins to provide high-frequency decoupling. Surface mounted types are recommended because of their lower lead inductance.

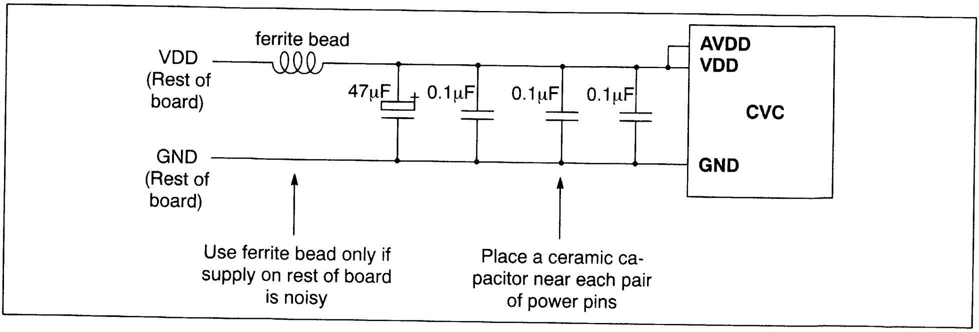
If (and only if) the power supply on the board is particularly noisy, the device can be isolated by the use of a local **VDD** plane. This should be connected to the global power plane by means of a ferrite bead, which will filter out high frequency noise on the global supply. The ferrite bead should have an impedance of around 100Ω at 100MHz. The local power plane should not be connected to a separate power supply since this incurs a risk of latch-up. If the use of a ferrite bead introduces too much inductance into the power supply to the device, observed as excessive variations of the potential of the local VDD plane, the ferrite bead may be replaced by one or more wire connections to the main VDD power plane.

Although the DACs draw their supply current through the **AVDD** pin, the **AVDD** supply is actually connected to the **VDD** supply via the chip substrate - a resistance of approximately 5 ohms. It is therefore essential that the **AVDD** and **VDD** pins are connected together on the board to eliminate any possibility of excessive substrate currents within the device.

A diagram of a suitable power supply arrangement is shown in Figure 5.1.

Another point to consider is that the power supplied to the actual die can be corrupted by the parasitic inductance inherent in the device package and leads. If the chip has to draw a sudden pulse of current, a back-e.m.f. is generated across this inductance which results in "ringing" on the internal supply rails. This effect cannot be eliminated, but it is helpful not to compound it. In order to obtain the best picture quality therefore, the device should be soldered directly onto the PCB and should not be socketed.

Figure 5.1. Suggested power supply arrangement


ferrite bead
VDD
(Rest of board)
47µF
+
0.1µF
0.1µF
0.1µF
AVDD
VDD
CVC
GND
GND
(Rest of board)
Use ferrite bead only if supply on rest of board is noisy
Place a ceramic capacitor near each pair of power pins

5.3 PHASE LOCKED LOOP POWER SUPPLY REQUIREMENTS

Why use a phase locked loop?

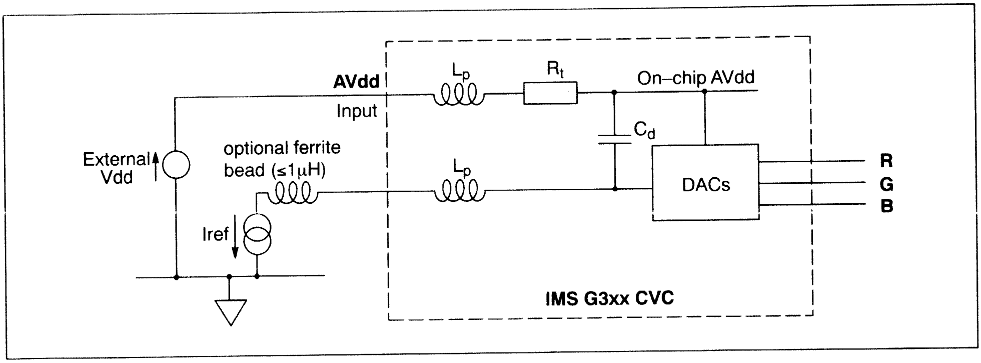
SGS-THOMSON CVCs employ an on-chip phase-locked loop (PLL) to multiply the relatively low-frequency input clock up to the pixel rate. This avoids routing a high-frequency clock around the board, reducing RFI and dependence on fast edge rates. An ECL compatible clock input also becomes unnecessary. The **ClockIn** should still be generated from a crystal oscillator for maximum temperature stability. The input clock edges must also be monotonic in the 10% to 90% range.

PLL power supply

A 1μF capacitor must be fitted between **CapPlus** and **CapMinus**. It must have sufficiently low e.s.l. (equivalent series inductance) and e.s.r. (equivalent series resistance) such that the total impedance connected between **CapPlus** and **CapMinus** across the frequency range 100KHz to 10MHz is less than 3Ω. The use of a ceramic capacitor is recommended because many tantalum capacitors have an e.s.r. greater than 3Ω.

The total PCB track length must be less than 50mm to minimize the resistance and inductive reactance in series with the capacitor. If this rule is obeyed then the capacitor e.s.r. will form nearly all of the impedance in the 100kHz - 10MHz range. It is then only necessary to ensure that the e.s.r. is less than 3Ω. The ideal arrangement is to use a surface-mounted capacitor inside the CVC package outline.

Figure 5.2. Equivalent circuit for on-chip AVDD supply



5.4 EXTERNAL CURRENT REFERENCE CIRCUITRY

The current drawn by the DACs from the **AVDD** pin is proportional to the DAC output levels. Because of the package inductance (L_p) and the on-chip resistance of the **AVDD** rail (R_t), the on-chip **AVDD** supply fluctuates during the picture scan time. See Figure 5.2.

The external current reference develops an on-chip bias voltage which must track the on-chip **AVDD** supply to ensure the correct DAC output level. The capacitor C_d (actually made up of the transistors in the DAC circuit) ensures that this is so, *provided* that the current source impedance is much higher than that of the capacitor across the DAC frequency range. In the time domain this means that the current source must have a fast response time, i.e. allow rapid voltage fluctuations across it whilst maintaining a constant current.

Current reference – decoupling

It may be clear from this that any attempt to "decouple" the **Iref** input by putting a capacitor across the current source will make the picture quality worse! The presence of such a capacitor will slow down the response of the current source, thus preventing the on-chip Vref from tracking the on-chip **VDD**.

Since capacitors are not to be used, other measures should be taken to prevent noise being injected into the **Iref** pin. The best method is to make the length of PCB track from the current source output (e.g. the LM334 **V+** pin, Figure 5.3(b)) to the device **Iref** pin less than 10mm in length. A ground track should also be placed either side of the **Iref** track to shield it from possible crosstalk. Use of a board with power planes is essential for the same reason. If more than two signal layers are present, digital signals *must not* cross the **Iref** track except where shielded by a power plane.

High frequency noise in the current reference may be further removed by introducing an inductance (value ≤1µH) or ferrite bead in series with the current reference (see Figure 5.8).

For systems where the power supply noise cannot be improved to the level where the current reference is able to track the noise on **AVDD**, a 100nF low inductance chip capacitor in parallel with a 47µF tantalum capacitor may be used to decouple the **Iref** input to AVDD. This is not recommended for high system performance as it prevents the current reference operating properly. This method also makes the value of the reference current dependent on the average output current supplied by the device.

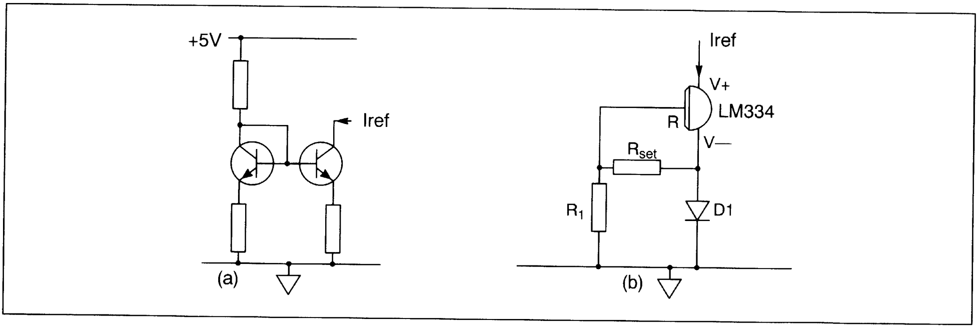
Current reference circuits

To ensure that the output current of the DACs is predictable and stable with temperature variations, an active current reference is essential. Figure 5.3 shows two designs of current reference.

Figure 5.3(a) shows the use of a transistor acting as a current reference. The value of **Iref** is set by the three resistors and power supply voltage. The thermal variations in the transistor's base emitter voltage are compensated for by the use of a second transistor acting as a forward biased diode.

Figure 5.3(b) shows the use of the LM334 precision current source as a current reference (in its temperature compensated configuration). The reference current is set by a single resistor (R_{set}) and is independent of the value of power supply voltage.

Figure 5.3. Current reference circuits



Calculating the Iref value

The value of Iref for IMS G3xx CVC devices depends on the video standard being generated and the type of termination. The method of calculating this value and an example, including the component values for Figure 5.3(b), is given below.

Iref is found from the formula:

$$\text{Iref} = \frac{120}{N} \times \frac{V_{peak}}{R_{out}}$$

R_{out} is the transmission line impedance in parallel with the source termination resistor. (See the section on video DACs). V_{peak} is the peak white voltage required while N is the number of current units switched on at the peak white level. This depends on whether the sync and blank pedestals are being used and is found from the following table:

	Color data (full scale)	Blanking pedestal	Sync
Units	255	20	108

The reference current can be provided by the LM334 precision current reference in the circuit of Figure 5.3(b).

The value of R_{set} determines the reference current supplied. Once the required Iref has been found, R_{set} is given by:

$$R_{set} = \frac{66.7\text{mV} \times 2}{\text{Iref}}$$

Diode D_1 provides temperature compensation - a 1N4148 or equivalent should be used. The ratio of R_1 to R_{set} determines the amount of temperature compensation - typically R_1 should be $10R_{set}$, but the optimum ratio should be found by experiment. See the LM334 datasheet for further details.

Example

An EIA-343-A compatible video output is required, driving a 75Ω doubly-terminated transmission line. Calculate the value of Iref and R_{set}.

The EIA-343-A standard uses composite sync and a blank level pedestal, so

$$N = 255 + 108 + 20 = 383$$

From the standard specification,

$$V_{peak} = 1.054\text{V}$$

The DAC output sees a 75Ω transmission line in parallel with a 75Ω termination resistor, because the system is doubly-terminated. So

$$R_{out} = 75\Omega || 75\Omega = 37.5\Omega$$

$$\Rightarrow \text{Iref} = \frac{120}{383} \times \frac{1.054}{37.5} = 8.81\text{mA}$$

R_{set} is then given by:

$$R_{set} = \frac{66.7\text{mV} \times 2}{8.81\text{mA}} = 15\Omega$$

R_1 will then be approximately 150Ω - a more accurate value can be found by experiment.

5.5 THE DAC OUTPUT CIRCUITRY

Double Termination

DAC outputs are driven at the pixel rate and hence should be treated with extreme care. Sluggish edges and/or noise on these outputs will degrade the picture quality.

The best edge rates are obtained when the RC time constant at the DAC output is minimized. It is for this reason that double termination is recommended: the impedance seen at the DAC output is then 37.5Ω (termination resistance in parallel with the transmission line impedance) instead of the 75Ω that would be present if single termination was used. As well as halving the RC time constant, this technique has the added bonus that reflections resulting from a mismatch at the monitor end (the cable and/or monitor input may not have exactly 75Ω impedance) will be neutralized when they return to the DAC output. Again better picture quality will result.

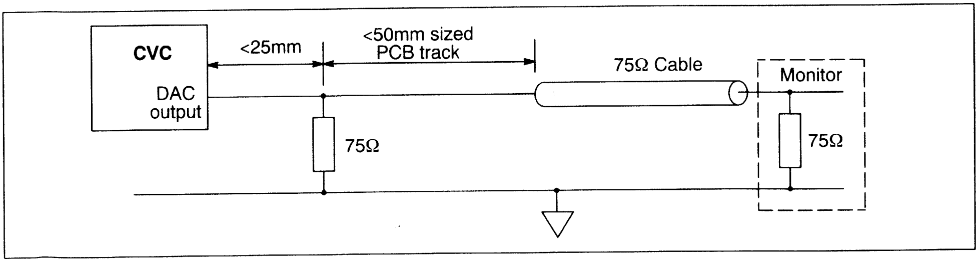
The PCB track from the DAC output pin to the termination resistor should be shorter than 25mm, in which case it can be treated as a lumped capacitance. This capacitance should be minimized to keep the RC time constant at the DAC output small. The best way to do this is to place surface-mounted resistors adjacent to the **Red**, **Green** and **Blue** pins just outside the device package outline.

The PCB tracks from the termination resistors to the 75Ω video connectors should be sized to form 75Ω transmission lines to minimize reflections, thus improving the DAC output waveform. Because the characteristic impedance of PCB tracks can only be guaranteed to within ±10%, these tracks should be made less than 50mm long so that their transmission line delay is short. Reflections due to mismatch will then occur mainly during the DAC output rise and fall times, so that settling time is not unduly affected. An added bonus of keeping the distance between the device and the video connectors short is that RFI is kept to a minimum.

The recommended arrangement is shown in Figure 5.4.

Note that a PCB with power planes MUST be used if transmission lines are to be formed from PCB tracks. The power planes will also shield the DAC outputs from logic signals on the opposite side of the board. Shielding from signals on the same side can be achieved by running ground tracks either side of each DAC output.

Figure 5.4. Double termination of DAC output

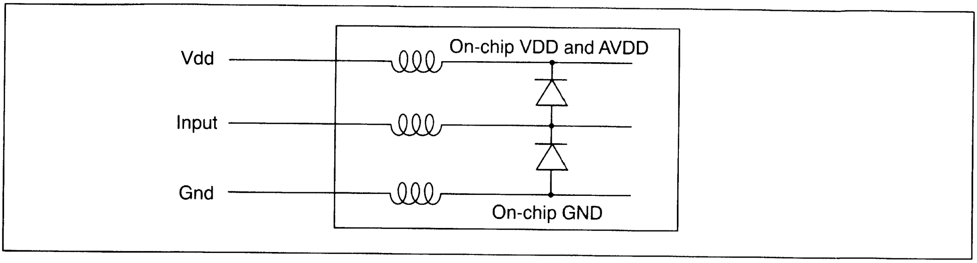


Output protection

CMOS devices are susceptible to damage from high electrostatic voltages. Normal antistatic precautions should be observed when handling them during system manufacture.

Once assembled into a system, devices are much less exposed to static damage. However if the analog outputs are made available at connectors to other equipment there is still a risk of over or under voltage behavior. Protection diodes to the power rails are recommended at the analog outputs for this reason. This is shown in Figure 5.8.

Figure 5.5. Equivalent circuit of esd protection structure



5.6 DIGITAL SIGNALS - AVOIDING OVERSHOOT, UNDERSHOOT AND OUTPUT DRIVE LIMITATIONS

It is important that all the inputs and outputs of the device obey the DC operating conditions specified in the product datasheet. Failure to comply with this will degrade the power supply to the device and the quality of the analog output. Care should also be taken to ensure that other digital signals on the board do not induce noise into sensitive analog areas.

Inputs

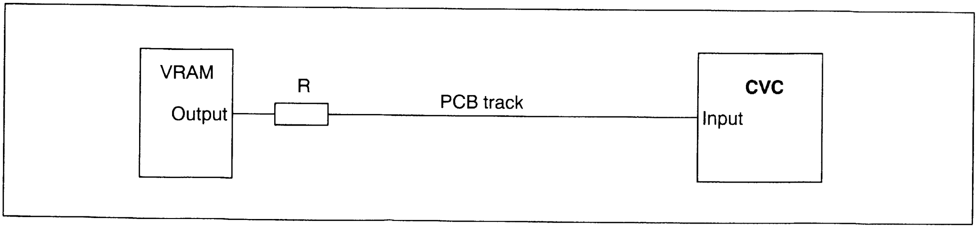
All inputs are protected from electrostatic discharge (esd) by circuits which appear very much like diodes. These are *not* intended to absorb transients caused by poor design and/or layout of the driving circuitry. The equivalent circuit of Figure 5.5 illustrates the problem.

If the input reaches VDD+0.5V, the diode to **VDD** will turn on. Further positive excursions of the input will start to drag the on-chip power supply in a positive direction. This will have a detrimental effect on the DAC circuitry in that noise will occur on the DAC outputs. Inputs which drop below –0.5V will not cause this effect because the DACs have their ground supply separated from that of the digital inputs. However, the reliability of the chip will be degraded. Therefore undershoot and overshoot greater than 0.5V outside the supply rails should be eliminated.

The most common occurrence of undershoot and overshoot is on the pixel inputs, caused by driver outputs not being matched to the PCB track leading to the device. The symptom is that of vertical lines appearing on the screen separated by a number of pixels equal to the multiplex ratio.

The easiest solution is to insert a series termination resistor at the driver output to match its impedance to that of the PCB track. See Figure 5.6.

Figure 5.6. Series termination of video RAM outputs



A suitable value for R should fall within the range 10Ω to 100Ω, depending on the PCB and output driver characteristics. It should be determined by experiment, i.e. R should be varied until the undershoot and overshoot occurring at the device inputs is within 0.5V of the power supplies.

Other advantages of using slower edge rates include less RFI and less risk of noise coupling into sensitive analog circuitry.

Outputs

The capacitive loading on each of the digital outputs *must* be kept below 25pF. The equivalent circuit of Figure 5.7 shows why.

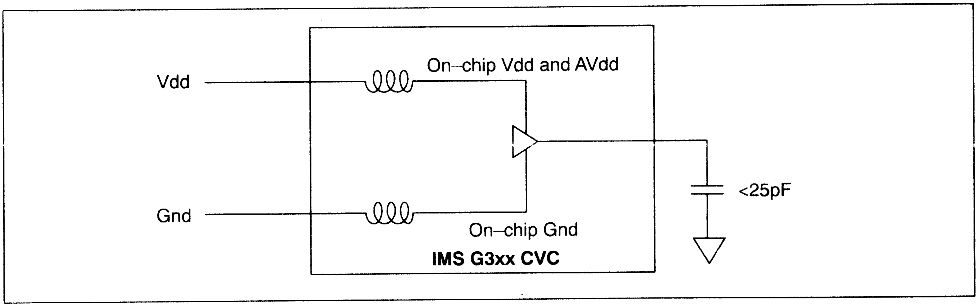
When an output driver switches, the capacitive load is charged or discharged via the **VDD** or **GND** supplies. A back-e.m.f. will then be generated across the parasitic inductance in the package, causing the on-chip power rails to bounce. Increasing the load to anything greater than 25pF will cause this bounce to be excessive, resulting in screen noise.

A common mistake is to load the **notSerialClk** and **notShiftClk** outputs excessively. Because these run at some fraction of the pixel rate (1/n), vertical stripes will appear on the screen separated by n pixels.

For the IMS G332/364 only, there is another requirement for the **notSerialClk** and **notShiftClk** loading which does not affect picture quality but does affect timing. These two outputs should be loaded identically to prevent skew between them. The presence of such skew might make the pixel setup and hold times difficult to achieve.

The problem of overshoot and undershoot can be minimized by using slower logic families to drive signals into the CVC. This has the advantage of reducing the edge rate of the signals without adding termination resistors. Faster logic families, such as 'F' series TTL, should only be used on preference to slower devices where there is no other way of satisfying timing specifications.

Figure 5.7. Loaded digital output driver



5.7 RADIATED POWER FROM GRAPHICS SYSTEMS

In order to comply with FCC and European regulations, computing systems, including their graphics components, must not emit Radio Frequency Interference beyond set limits. The following guidelines are intended to help designers minimize the RFI emitted from the video components of a graphics system.

All conductors carrying time varying currents radiate electromagnetic radiation. This radiation may in turn induce currents in other nearby conductors. Radiated power from a conductor increases when it is physically large and is separated from a ground plane by a large distance. That is to say, good radiating aerials are large open systems. In a typical graphics board, the major radiating aerials are the tracks and wires surrounding the video components which carry the high frequency video signals.

To minimize the radiated power from a board a number of basic rules can be applied:

1 Keep tracks leading to and from the CVC short, particularly the pixel address, pixel clock and DAC outputs.

2 Always use at least a four-layer board with VDD and GND planes.

3 Avoid flying leads carrying high frequency signals which are not screened.

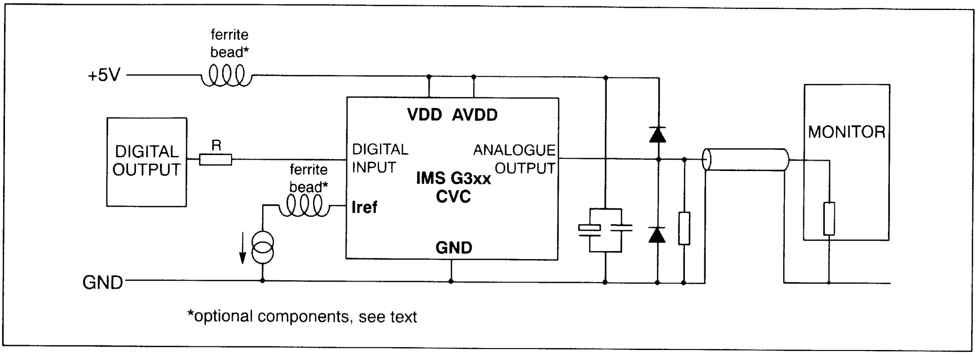
4 Place load resistors as close as possible to the analog outputs.

5 Major current transients occur when the DACs are switched on and off. As this current is drawn from AVDD and VDD, it is important to ensure that the power supply is efficiently decoupled to GND close to the device for low and high frequencies. This can be particularly important if a ferrite bead has been used to isolate a local VDD plane for the device from the main VDD power plane.

Radiated power increases with the frequency of the radiating signal (to the fourth power). It is therefore desirable to have as few signals as possible with very fast edge rates since these will have high frequency components. As mentioned earlier, the use of slower logic families and series termination of powerfully driven signals will reduce the risk of excessively fast edges.

In summary, to minimize RFI emissions, keep all tracks carrying video and near-video rate signals as short as possible, keep edge rates as slow as possible and ensure good power supply coupling around all the video circuitry.

Figure 5.8. Suggested circuit



*optional components, see text

5.8 SUMMARY

Figure 5.8 shows a suggested circuit summarizing some of the techniques described earlier in this appendix. These include power supply decoupling, series termination of a digital output, decoupling of the current reference and double termination of an analog output.

Summary – DOs and DON'Ts

DOs

- Do use a board with power planes
- Do short **VDD** and **AVDD** together
- Do use 47μF tantalum and 0.1μF capacitors for supply decoupling
- Do solder the device directly into the PCB or use zero-profile socket pins
- Do keep the **V+** pin of the LM334 within 10mm of the device **Iref** pin
- Do shield the track to **Iref** with ground tracks either side
- Do use double termination from the analog outputs to the monitor
- Do keep termination resistors within 25mm of DAC outputs - surface mounted resistors inside the device package outline are recommended
- Do size PCB track from termination resistor to video connector such that it forms a 75Ω transmission line less than 50mm long
- Do keep DAC outputs away from logic signals - preferably shield with ground tracks

DON'Ts

- Don't use separate supplies for **VDD** and **AVDD**
- Don't put the CVC in an IC socket
- Don't put unnecessary capacitors across the current source
- Don't allow digital inputs to transgress the supply rails by more than 0.5V
- Don't make edge rates of logic signals any faster than necessary

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

COMPACT DESIGN CONSIDERATIONS WHEN USING THE IMS G335

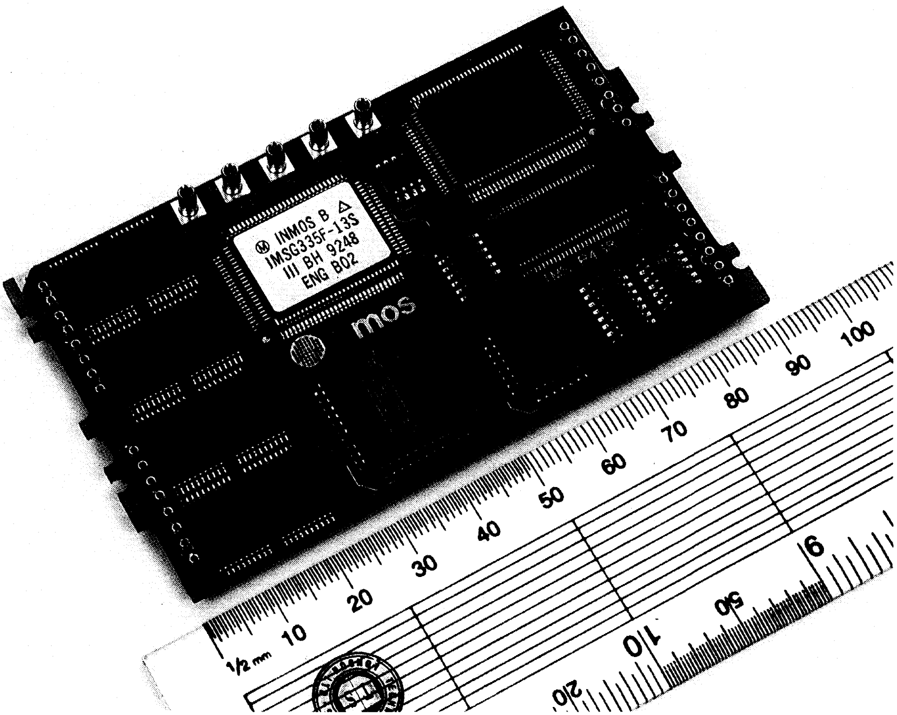
GRAPHICS APPLICATIONS GROUP, BRISTOL

6.1 INTRODUCTION

A complete microprocessor based graphics subsystem has been designed using an SGS-THOMSON IMS G335 Color Video Controller (CVC). This graphics board design focuses upon two basic requirements; simplicity and compactness in order to accommodate the graphics subsystem in a 3.3in.×2.15in. board area (see Figure 6.1). At the heart of the design is a 30MHz IMS T805 transputer which has a number of components mapped into address space using fast Programmable Logic Devices (PLDs). The graphics interface to the display is controlled by the IMS G335 which can be programmed to drive a wide range of monitors. To allow execution of 'local' graphics based programs, 4Mbytes of 133ns cycle dynamic memory (DRAM) is provided. The screen store is then resident within 2Mbytes of fast dual-ported Video RAM (VRAM) which provides bandwidth efficient access to the memory from both the transputer, for graphics drawing operations, and from the IMS G335 CVC in order to display images stored in the memory.

This document outlines a minimal design suitable for compact graphics products based upon the IMS G335. Important factors such as reducing interface logic, choice of package and board layout considerations are discussed in addition to a description of the functional specification and the resulting design methodology employed in the layout and manufacture of the board.

Figure 6.1. IMS B438 integrated graphics TRAM



6.2 EXTERNAL MEMORY INTERFACING

The transputer's External Memory Interface (EMI) configuration is defined using two PLDs. The memory configuration:

- supports two possible cycle lengths:

  DRAM is accessed with no wait states

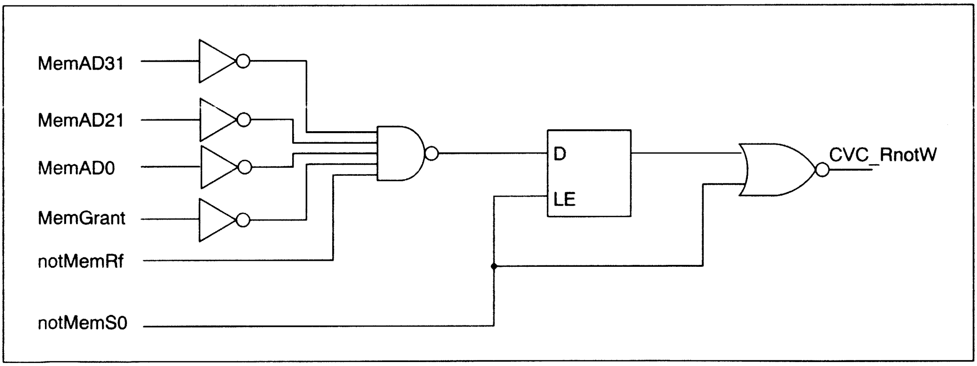
  VRAM is accessed with a single wait state (through sampling the notMemS4 signal)
- extends the EMI memory cycle state T2 to adhere to the address hold time of the CVC's multiplexed address/data bus.
- adds a further T-state at the end of a cycle to the CVC to adhere to the **notChipSelect** minimum inactive time.

The selected configuration provides strobe signals **notMemS(0..4)** which are suitable for interfacing to the selected dynamic memories and the IMS G335. In order to support address decoding from the transputer's multiplexed address/data bus, an ALTERA EPM7032 EPLD is used. This device allows decode signals to be generated and independently latched during the address phase of the transputer's EMI cycle, using the falling edge of **notMemS0**. Figure 6.2 illustrates a portion of the schematic entry of the ALTERA EPM7032. The example illustrates how the IMS G335 is partially decoded between 00000000h and 00FFFFFFh in the transputer's memory map to generate the CVC_RnotW signal.

Signals such as 'DRAM_Cycle' are generated by latching the decode addresses (MemAD31, etc.), the transputer's DMA acknowledge signal **MemGranted**, and the **notRefresh**, signal indicating whether the transputer is performing a refresh cycle.

A second PLD is used to generate RAS and CAS strobes to the VRAM, such that the transputer's EMI strobes control bus cycles when the CVC is sampling pixels from the VRAM SAMs, and a state machine controls bus cycles for VRAM read transfers to the SAM.

Figure 6.2. Control signals to the IMS G335



6.2.1 ASSIGNING MULTIPLEXED ADDRESSES

Multiplexed addresses presented to the dynamic memories are arranged to allow the use of a single 74ACT16841. There are three instances of addressing which need to be defined and minimized accordingly:

- Transputer access to DRAM
- Transputer access to VRAM
- CVC's access to VRAM (for SAM transfer cycles)

A second consideration is the application of row and column addresses to the different memory architectures. The 1024K×4 DRAMs require ten row and column addresses. The 256K×8 VRAMs require nine row and column addresses, but are interleaved on word boundaries using the least significant address bit, to support 15 and 16bpp modes on the IMS G335

A fundamental addressing requirement in graphics systems is to ensure that the processor accesses VRAM column locations least significantly and row locations most significantly. This ensures that processor read or write cycles are performed across *rows* which are then mapped onto the screen by *row* transfers. In this design **A(3..11)** decode the column locations of the VRAM array and **A(12..20)** decode the row locations. The interleaved memory mode addressing requirements with the IMS G335 also dictate that the processor must access the two VRAM banks alternately on word boundaries. This is done by decoding address **A2**.

These requirements can be satisfied by the row and column address assignments shown in Table 6.1.

In order to allow the IMS G335's transfer address presentation to use the same latch and multiplexor hardware (to avoid swapping of row and column addresses) the IMS G335 provides increment control bits in the control register which *must be set to an increment of 1024 (or 512 in designs implementing split SAM VRAMs)*. This provides a count sequence on the row address as required by the VRAM during row transfer cycles.

Table 6.1. Row and column address assignments

AddrBus	MuxAddrBus	T805 → video RAM	G335 → video RAM	T805 → DRAM
A21	MA9			RA9
A20	MA8	RA8	RA8	RA8
A19	MA7	RA7	RA7	RA7
A18	MA6	RA6	RA6	RA6
A17	MA5	RA5	RA5	RA5
A16	MA4	RA4	RA4	RA4
A15	MA3	RA3	RA3	RA3
A14	MA2	RA2	RA2	RA2
A13	MA1	RA1	RA1	RA1
A12	MA0	RA0	RA0	RA0
A11	MA8	CA8	CA8	CA8
A10	MA7	CA7	CA7	CA7
A9	MA6	CA6	CA6	CA6
A8	MA5	CA5	CA5	CA5
A7	MA4	CA4	CA4	CA4
A6	MA3	CA3	CA3	CA3
A5	MA2	CA2	CA2	CA2
A4	MA1	CA1	CA1	CA1
A3	MA0	CA0	CA0	CA0
A2	MA9	Bank select bit		CA9

6.2.2 LIMITATIONS OF THE CAS FIELD BITS

The architecture of the CVC family limits the use of scrolling functionality such that scrolling is not possible when the device operates with seamless midline updates. This particular function is only possible when the VRAM shift register length maps onto a single line. It is thus related to the number of bits per pixel. This design implements VRAMs with a 512×512×8 architecture. When running in interleaved mode this configuration allows access to a serial port with a 512×32 architecture. A 16bpp option is a typical color resolution which will map onto a single 1024 pixel line to allow scrolling, since 1024×16-bit pixels fill the serial port. In this mode the TopOfScreen register is incremented to allow scrolling with an increment of the CAS field bits, which is given by *number of pixels/notShiftClk period*.

The minimum increment in the CAS field is typically a value of 4 pixels (equivalent to 1SClk) but it must be doubled to 8 in this design since A3 is the least significant address.

6.3 THE IMS G335 COLOR VIDEO CONTROLLER

The IMS G3xx series of color video controllers are designed with 4:1 pixel acceleration, allowing the serial ports of the VRAMs to be clocked at a quarter of the video rate in 8bpp pseudo color mode. The **notShiftClk** output from the CVC is designed to run at rates greater than one quarter of the video frequency. In order to maintain high video rates at 16bpp an interleaved mode is provided which allows two pixels to be input on one edge of **notShiftClk** and another two on the second edge. To maintain a specified clock rate on the serial port of the VRAMs two banks must be used and driven in antiphase. This maintains a shift rate from each VRAM bank at a rate of one quarter of the clock rate.

The choice of how to implement the bank multiplexer depends on the maximum video dot rate that the system is designed to run at. Because of the small board estate of this design, the fast pixel multiplexors (usually required to support pixel interleaving with the IMS G335) were omitted and traded off against maximum pixel frequency. Since the VRAM $\overline{\text{SOE}}$ turn on/off timings are sufficiently fast, it is possible to multiplex between the two banks *without using any external multiplexing hardware*. Such a system is shown in Figure 6.3, with its associated timing diagram in Figure 6.4.

Using worst case propagation delays specified for the 74F04 inverters, and with Micron 2Mbit VRAMs (serial output turn on time of 12ns), the system will run at up to 95MHz video rate, allowing a 1280×1024 screen at 8bpp or 1024×768 screen at 16bpp with the IMS G335 at around 85-90Hz frame rate. This is verified by summing t_{PD} (74F04), t_{SEA} (MT42C8256) and the IMS G335 pixel setup time t_{PVLH}. This is 6+12+2.5 = 20.5ns. This means that the VRAM shift clock period must be 41ns minimum. The maximum video rate is then 4/41ns = 95MHz at 16bpp or 47.5MHz at 24bpp using 2:1 interleaving.

Figure 6.3. Interleaving using VRAM $\overline{\text{SOE}}$

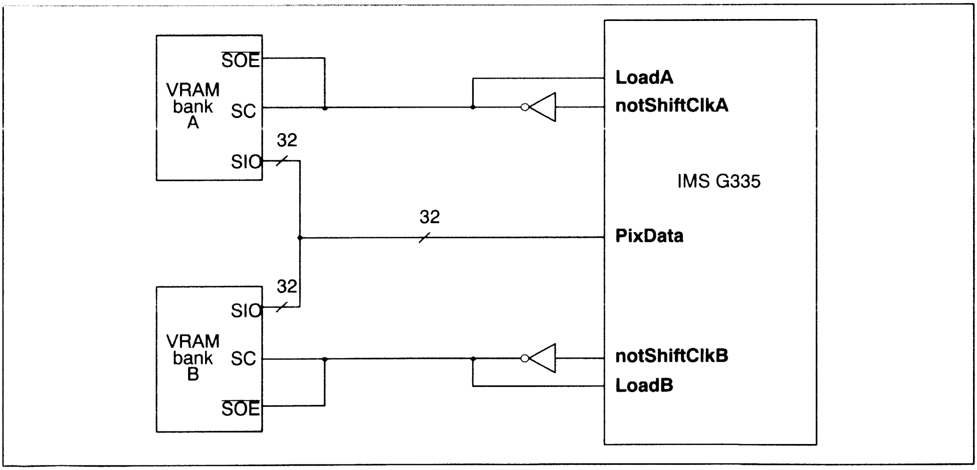


Figure 6.4. $\overline{\text{SOE}}$ interleaved timing diagram for 95MHz 16bit color screen refresh using the IMS G335

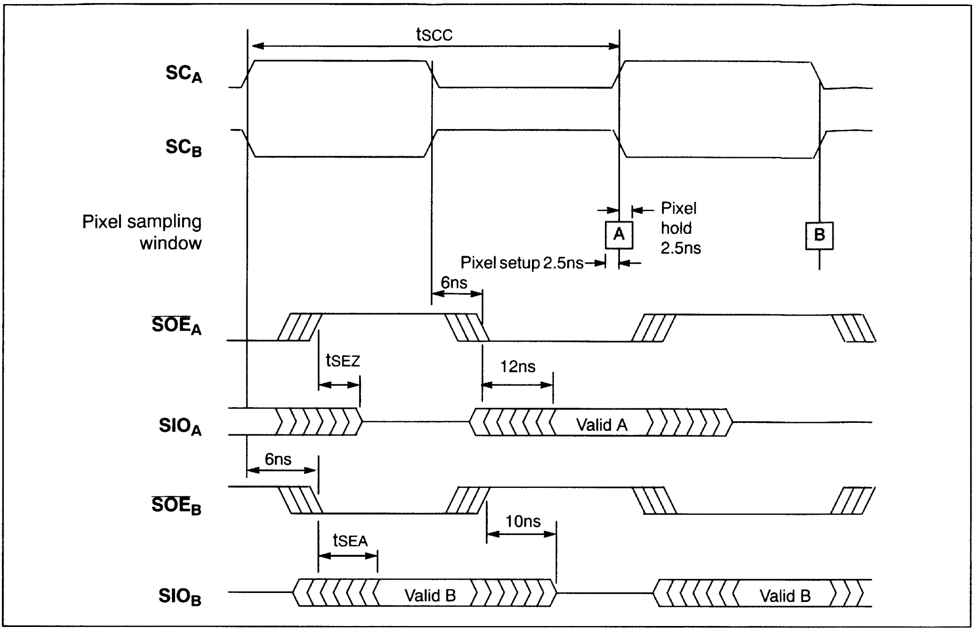


Table 6.2.

Symbol	Description	Min.	Max.	Unit	Notes
tSCC	**SC** clock		41	ns	
tSEA	Access time from Serial Enable (SE)	–	12	ns	1
tSEZ	Serial output buffer turn-off delay from SE	3	10	ns	1

NOTE:
1 These timings apply to 2Mbit Micron VRAM devices

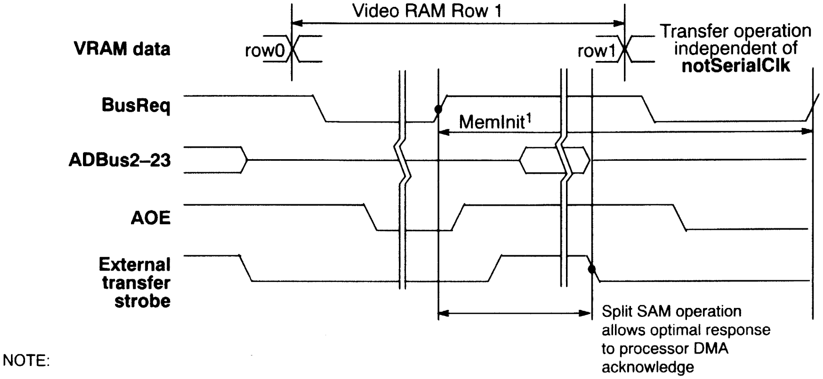
From Figure 6.3, it is apparent that, in addition to assessing the ability to meet the pixel setup time of 2.5ns (when the maximum buffer propagation delay and VRAM output turn-on time are considered), maintaining the 2.5ns pixel hold time is also important. The pixel hold time is determined by the minimum propagation delay of the 240/241 buffers added to the time it takes the VRAM serial outputs to turn on/off in response to $\overline{\text{SOE}}$ changing. Minimum delays for the buffers are quoted as 2ns and the minimum serial output turn-off time as 3ns. This therefore ensures that the pixel hold time will be met

6.4 SPLIT SAM TRANSFERS WITH THE IMS G335

The split SAM functionality provided by the IMS G335 provides a number of benefits in a transputer based graphics subsystem. The transputer, because of its flexible EMI, has a high maximum DMA latency. Non-split SAM systems implementing a packed framestore must be able to exactly synchronize the moment of VRAM transfer to the point when data from SAM is just about to be exhausted. In such systems the CVC must retain control of the bus for long enough to allow for worst case DMA latency.

In split SAM systems the exact moment of transfer is non-critical and the G335 can begin a transfer as soon as the processor releases the bus to the CVC. The transfer point is variable, depending on the activity of the transputer's EMI and is not dependent on a worst case value programmed into the CVC. The CVC can schedule the transfer at any time within a duration equal to half the SAM length.

Figure 6.5. Split SAM allows optimal response to DMA acknowledgement


Video RAM Row 1
VRAM data
rowO
row1
Transfer operation independent of **notSerialClk**
BusReq
MemInit¹
ADBus2–23
AOE
External transfer strobe
Split SAM operation allows optimal response to processor DMA acknowledge

NOTE:

1 MemInit is a programmable G335 framestore description parameter. In split SAM mode the G335 uses MemInit to set the period of transfer requests.

6.5 PHYSICAL DESIGN CONSIDERATIONS

The highly compact feature size of this design is maintained not only through design functionality optimization but by exploiting surface mount technology. This technology provides a number of benefits to such a requirement:

- Reduced footprint of components increasing board density
- Reduced board costs
- Localized routing in smaller surface areas
- Double sided component population

Although extensive use of surface mount components may decrease available via space on a multilayer p.c.b., placement of components can be optimized to limit this lost surface area by exactly duplicating the component footprint on both surfaces of the board. This practice allows high placement packing while retaining adequate routing paths through via holes from one layer to another.

Eight 1024K×4 DRAM devices are used to build a 4Mbyte memory array. The devices are packaged in 28 pin TSOPII packages. Eight 512K×8 VRAM devices support two 1MByte VRAM framestores which are interleaved on word boundaries. These components are available in a 40/44-pin TSOP(R–5) package with standard and reverse pin out options simplifying the routing requirements.

Other package selection and placement points are outlined as follows:

- A 100-pin CQFP transputer is used to allow the under surface to be used for discrete link support, speed select, decoupling and PLL components.
- 5 SMR connectors are used since they have approximately a 40% reduced width and length, contributing to the footprint, compared to the more popular SMB connectors
- Locality of VRAM banks allows efficient local routing
- With an optimized placement, the design rules are set such that routing on surface layers is predominant.
- Transputer placement optimizes the **ADBus** interface to the IMS G335 and the inputs to the address multiplexor and is in close proximity with the TRAM connectors link, subsystem and 5MHz clock pins.

6.6 CONCLUSION

This note illustrates that the IMS G335 can provide a highly integrated and flexible solution to high resolution pseudo and true color graphics systems, within a small compact board design based on the IMS T805 transputer.

The design supports 135MHz operation at 8bpp and up to 95MHz at 16bpp, yet is accommodated on a p.c.b. with an area similar to that of a credit card.

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

CVC PARAMETER CALCULATIONS FOR DRIVING STANDARD PC DISPLAYS

GRAPHICS APPLICATIONS GROUP, BRISTOL

7.1 INTRODUCTION

The IMS G3xx Color Video Controllers (CVCs) are a family of devices integrating all the functions required for controlling high resolution color graphics displays. These extremely flexible devices have a simple address-mapped microprocessor interface and highly programmable Video Timing Generator (VTG) on-chip. The programmable VTG enables a wide variety of monitors and TVs to be driven at almost any resolution and frame rate, including PC monitors displaying VGA and SVGA standard resolutions.

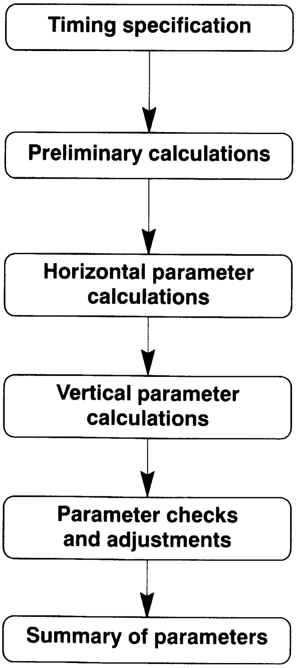
This note outlines the simple calculation of display parameters and thus the VTG register values required to drive these standard PC displays with the IMS G3xx family of Color Video Controllers (640×480 at 25MHz, 800×600 at 35 and 36MHz, 1024×768 at 65MHz, and 1280×1024 at 105MHz). All calculations have been included, as well as relevant decisions being discussed, for completeness and instruction.

The parameter calculations in this note apply only to the IMS G332, G364 and G335. The IMS G300 requires certain parameters to be calculated differently to those described here. This product has now been obsoleted and is **not** recommended for new designs. The IMS G335 now offers the best cost/ performance solution and is recommended for all new designs.

7.1.1 DERIVATION OF PARAMETERS

The format shown in Figure 7.1 is used throughout this note to show the derivation of parameters for each display type.

Figure 7.1. Derivation of parameters


Timing specification
Preliminary calculations
Horizontal parameter calculations
Vertical parameter calculations
Parameter checks and adjustments
Summary of parameters

These are timing specifications obtained from monitor handbooks. As such they may vary slightly (but insignificantly) from the VESA standards. The specifications comprise horizontal and vertical frequencies together with the timings shown in Figure 7.2.

These calculations derive the video frequency from the horizontal display period and horizontal resolution. Parameters are calculated for the nearest integer video frequency to that determined theoretically, and, where different, for the nearest or most appropriate multiple of 5MHz (a commonly used input clock frequency for CVCs and transputers).

These parameters define the components of the horizontal timing waveform in Screen Units (one Screen Unit = 1 notSerialClk period).

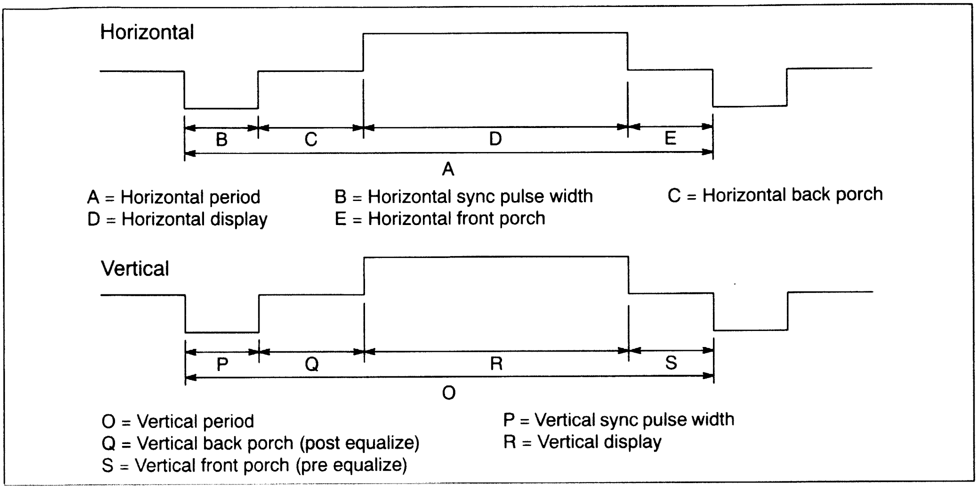
These parameters define the components of the vertical timing waveform in Half Linetime periods.

The datasheets for IMS G3xx devices outline a number of restrictions on parameter values which must be observed if the VTG is to function correctly. The calculated parameters are verified against these restrictions.

7.1.2 IDENTIFICATION OF TIMING PERIODS

Timing specification periods refer to the following horizontal and vertical timing waveforms:

Figure 7.2.



A = Horizontal period
D = Horizontal display
B = Horizontal sync pulse width
E = Horizontal front porch
C = Horizontal back porch

O = Vertical period
Q = Vertical back porch (post equalize)
S = Vertical front porch (pre equalize)
P = Vertical sync pulse width
R = Vertical display

7.2 CALCULATIONS FOR 640×480 (VGA)

7.2.1 TIMING SPECIFICATION

Horizontal						Vertical					
		Units			**Units**			**Units**			**Units**
Freq.	31.5	KHz	C	1.89	µs	Freq.	60	Hz	Q	1.02	ms
A	31.77	µs	D	25.17	µs	O	16.68	ms	R	15.246	ms
B	3.77	µs	E	0.94	µs	P	0.064	ms	S	0.35	ms
Misc: Separate digital syncs (both negative active), no sync on green.											

7.2.2 PARAMETER CALCULATIONS

Preliminary calculations **640×480 at 25MHz (VGA)**

Given that:

- horizontal display = 25.17µs
- horizontal resolution = 640
- video frequency = 640 / 25.17µs = 25.427MHz

A video frequency of 25MHz will be assumed.

Horizontal parameters **640×480 at 25MHz (VGA)**

1 period notSerialClock (1 Screen Unit) = 4 / 25MHz = 160ns.

HalfSync = (B / notSerialClock) / 2 = (3.77µs / 160ns) / 2 = 11.78

Rounding to nearest integer, HalfSync = 12.
This gives a horizontal sync pulse width of (HalfSync × notSerialClock) × 2, hence horizontal sync = 3.84µs, which compares to the specification of 3.77µs.

BackPorch = C / notSerialClock = 1.89µs / 160ns = 11.8

Rounding to nearest integer BackPorch = 12.
This gives a horizontal back porch of BackPorch×notSerialClock = 1.92µs, which compares to the specification of 1.89µs.

Display = horizontal resolution / 4 = 640 / 4 = 160

This gives a horizontal display period of 160×notSerialClock = 25.6µs, which compares to the specification of 25.17µs.

Linetime = A / notSerialClock = 31.77µs / 160ns = 198.56

Rounding to nearest even integer, Linetime = 198.
This gives a horizontal period of Linetime × notSerialClock = 198×160ns = 31.68µs (31.56KHz), which compares to 31.77µs.

FrontPorch does not have to be explicitly set in the IMS G3xx products, but will be automatically set to:

FrontPorch = Linetime – ((2×HalfSync) + BackPorch + Display)

FrontPorch = 198 – ((2×12) + 12 + 160) = 2

This gives a horizontal front porch period of FrontPorch×notSerialClock = 2 × 160ns = 0.32µs

ShortDisplay < (Linetime / 2) – ((2 × HalfSync) + BackPorch)

ShortDisplay < (198 / 2) – ((2×12) + 12) < 99 – 36 < 63
Therefore ShortDisplay = 62

This is not specified in the specification, but must be programmed into the G3xx.

BroadPulse = (Linetime / 2) – (2 × HalfSync)

BroadPulse = (198 / 2) – (2 × 12) = 99 – 24 = 75

This is not specified in the specification, but must be programmed into the G3xx.

Horizontal frequency

Using these parameters, the horizontal period and thus frequency will be:
Horizontal frequency = 1 / horizontal period
h period = Linetime×notSerialClock = 198×160ns = 31.68µs.
So horizontal frequency = 31.56KHz.

Vertical parameters **640×480 at 25MHz (VGA)**

One period HalfLinetime = 31.68µs / 2 = 15.84µs.

Note that the period HalfLinetime is not a parameter or register set in the G3xx devices, but a period used in the calculation of the vertical parameters.

VSync = P / HalfLinetime = 0.064ms / 15.84µs = 4.04.

Rounding to nearest integer, VSync = 4. This gives a vertical sync pulse width of VSync × HalfLinetime = 4 × 15.84µs = 0.063ms.

VPre and VPost

VPre = S / HalfLinetime = 0.35ms / 15.84µs = 22.09

Rounding to the nearest integer, VPre = 22. This gives a VPre period of 0.348ms which compares to the specification of 0.35ms.

VPost can be set to VSync, as the specification sets the sum of VPost and VBlank rather than setting them individually.

VDisplay = vertical resolution × 2 (in halflines) so VDisplay = 960

This gives a vertical display period = VDisplay×HalfLinetime = 960×15.84µs = 15.206ms.

VBlank = (O / HalfLineTIme) – (VSync + VPre + VPost + VDisplay)

VBlank = (16.68ms / 15.84µs) – (4 + 22 + 4 + 960) = 63.03

Rounding this to the nearest even integer, VBlank = 64. This gives a vertical blanking time of VBlank × HalfLinetime = 64 × 15.84µs = 1.01ms.

Vertical frequency

The vertical frequency obtained using the vertical parameters is thus:

Vertical frequency = 1 / vertical period

Vertical period = (VSync+VPre+VPost+VDisplay+VBlank) × HalfLinetime

Vertical period = 1054 × 15.84µs = 16.69ms.

Thus vertical frequency = 1 / 16.69ms = 59.92Hz.

continued

7.2.3 PARAMETER CHECKS FOR 640×480 AT 25MHz

1 All parameters are non-zero.

2 Linetime must be even.

3 Half line point: Ha > (Linetime / 2) > Hb

where:

Ha = (2×HalfSync) + Backporch + Display

Hb = (2×HalfSync) + BackPorch

So:

Ha = (2 × 11) + 12 + 160 = 194

Hb = (2 × 11) + 12 = 34

So 194 > 80 > 34 hence this condition is met.

4 Total of vertical lines per frame must be integer, i.e. number of half lines must be even:

Number of halflines per frame = VPre + VSync + VPost + VDisplay + VBlank = 22 + 4 + 4 + 960 + 64 = 1054 = even.

5 For IMS G332/364:

BackPorch > (TransferDelay + 1)

for IMS G335:

BackPorch > (TransferDelay + 2)

TransferDelay is implementation dependent, and has not been calculated here.

6 TransferDelay ≤ Short Display

TransferDelay is implementation dependent, and has not been calculated here.

7 With the G332 and G364, Backporch must exceed 16. This condition has not been met, as BackPorch has been calculated to be 12. We cannot increase BackPorch by reducing FrontPorch, as FrontPorch is already only 2. We can reduce HalfSync from 12 to 9, which allows BackPorch to be increased from 12 to 18 (twice 3).

8 With the G335, BackPorch can be set to 12, but if BackPorch is set to below 16, then to set the cursor x position to be X, the cursor x position must be set to X + d, where:

d = (16 – BackPorch) × 4

7.2.4 SUMMARY OF G3xx PARAMETERS FOR 640×480 AT 25MHz

Register	G332 / G364	G335
HalfSync	9	12
BackPorch	18	12
Display	160	160
Linetime	198	198
ShortDisplay	62	62
BroadPulse	75	75
VPre	22	22
VSync	4	4
VPost	4	4
VBlank	64	64
VDisplay	960	960
(FrontPorch)	2	2

Note that for the G335 to set the cursor x position to be X, the cursor x position must be set to X + d, where:

d = (16 – BackPorch) × 4

7.3 CALCULATIONS FOR 800×600

7.3.1 TIMING SPECIFICATION

Horizontal						Vertical					
		Units			Units			Units			Units
Freq.	35.16	KHz	C	3.56	μs	Freq.	56	Hz	Q	0.6	ms
A	28.44	μs	D	22.22	μs	O	17.78	ms	R	17.07	ms
B	2	μs	E	0.67	μs	P	0.06	ms	S	0.03	ms
Misc: Separate syncs											

Note that 1 / 56Hz = 17.86ms, not 17.78ms as in the specification. Therefore 17.86ms will be used for O. The period S will also be set to be (1 / 56) – (P + Q + R).

7.3.2 PARAMETER CALCULATIONS

Preliminary calculations **800×600**

Given that:

- horizontal display = 22.22μs
- horizontal resolution = 800
- video frequency = 800 / 22.22μs = 36MHz

Either 36MHz or 35MHz (being a multiple of 5MHz) could be used. These are dealt with separately below.

Horizontal parameters **800×600 at 35MHz**

1 period notSerialClock (1 Screen Unit) = 4 / 35MHz = 114.29ns

Display = horizontal resolution / 4 = 800 / 4 = 200

This gives a horizontal display period of 200 × notSerialClock = 22.86μs, which compares to 22.22μs. The period 22.86μs must therefore be used for D in subsequent horizontal calculations.

Linetime = A / notSerialClock = 28.44μs / 114.29ns = 248.85

Rounding to nearest even integer, Linetime = 248. This gives a horizontal period of Linetime × notSerialClock = 248 × 114.29ns = 28.34μs (which corresponds to 35.28KHz). The period 28.34μs must therefore be used for A in subsequent horizontal calculations.

HalfSync = (B / notSerialClock) / 2 = (2μs / 114.29ns) / 2 = 8.75

Rounding to the nearest integer, HalfSync = 9. This gives a horizontal sync pulse width of (HalfSync × notSerialClock) × 2, hence horizontal sync = 2.06μs.

BackPorch = C / notSerialClock = 3.56μs / 114.29ns = 31.15

Rounding to nearest integer BackPorch = 31. This gives a horizontal back porch of BackPorch × notSerialClock = 3.54μs.

continued

Horizontal parameters (continued) **800×600 at 35MHz**

FrontPorch does not have to be explicitly set in the IMS G3xx products, but will be automatically set to:

FrontPorch = Linetime – ((2 × HalfSync) + BackPorch + Display)

FrontPorch = 248 – ((2 × 9) + 31 + 200) = –1

Now because FrontPorch is –1, either BackPorch, HalfSync or LineTime must be reduced to make it positive. BackPorch is already a little lower than the standard, and as the horizontal sync pulse width is a little higher than the standard, HalfSync may be reduced from 9 to 8. LineTime can be increased to 250; the period 28.57μS must therefore be used for A in subsequent horizontal calculations. This increases FrontPorch by 4×1 so FrontPorch = 3. This gives a horizontal front porch period of FrontPorch × notSerialClock = 3 × 114.29ns = 0.343μs.

ShortDisplay < (Linetime / 2) – ((2 × HalfSync) + BackPorch)

ShortDisplay < (250 / 2) – ((2×8) + 31) < 125 – 47 < 78
Therefore ShortDisplay = 77

This is not specified in the specification, but must be programmed into the G3xx.

BroadPulse = (Linetime / 2) – (2 × HalfSync) BroadPulse = (248 / 2) – (2 × 8) = 124 – 16 = 108.

This is not specified in the specification, but must be programmed into the G3xx.

Horizontal frequency

Using these parameters, the horizontal period and thus frequency will be:

Horizontal frequency = 1 / horizontal period

h period = Linetime × notSerialClock

h period = 248 × 114.29ns = 28.34μs

Horizontal frequency = 35.28KHz

Vertical parameters **800×600 at 35MHz**

One period HalfLinetime = 28.34μs / 2 = 14.17μs.

Note that the period HalfLinetime is not a parameter or register set in the G3xx devices, but a period used in the calculation of the vertical parameters.

R = vertical resolution × Linetime × notSerialClock

R = 600 × 248 × 114.29ns = 17.01ms.

Setting the period S to be (1 / 56) – (P + Q + R):

S = (1 / 56) – (0.06 + 0.6 + 17.01)ms = 0.187ms.

continued

Vertical parameters (continued) **800×600 at 35MHz**

VSync = P / HalfLinetime = 0.06ms / 14.17µs = 4.23.

Rounding to nearest integer, VSync = 4. This gives a vertical sync pulse width of VSync × HalfLinetime = 4 × 14.17µs = 0.057ms.

VPre and VPost

VPre = S / HalfLinetime = 0.187ms / 14.17µs = 13.2

Rounding to the nearest even integer, VPre = 14. This gives a VPre period of 0.198ms.

VPost can be set to the same as VSync, as it is not explicitly defined in the standard, so set VPost = 4. This gives a VPost period of 0.057ms.

VDisplay = vertical resolution × 2 (in halflines) so VDisplay = 1200.

This gives a vertical display period = VDisplay × HalfLinetime = 1200 × 14.17µs = 17.01ms.

VBlank = (O / HalfLineTIme) – (VSync + VPre + VPost + VDisplay)

VBlank = (17.86ms / 14.17µs) – (4 + 14 + 4 + 1200) = 38.41.

Rounding this to the nearest even integer, VBlank = 38. This gives a vertical blanking time of VBlank × HalfLinetime = 38 × 14.17µs = 0.538ms.

Vertical frequency

The vertical frequency obtained using these vertical parameters is thus:

Vertical frequency = 1 / vertical period

Vertical period = (VSync+VPre+VPost+VDisplay+VBlank)×HalfLinetime

Vertical period = 1260 × 14.17µs = 17.85ms.

Vertical frequency = 1 / 17.85ms = 56.01Hz

continued

7.3.3 PARAMETER CHECKS FOR 800×600 AT 35MHz

1 All parameters are non-zero.

2 Linetime must be even.

3 Half line point:

Ha > (Linetime / 2) > Hb

where:

Ha = (2 × HalfSync) + Backporch + Display

Hb = (2 × HalfSync) + BackPorch

So:

Ha = (2 × 8) + 31 + 200 = 247

Hb = (2 × 8) + 31 = 47

So 247 > 124 > 47 so this condition is met.

4 Total of vertical lines per frame must be integer, i.e. number of half lines must be even:

Number of halflines per frame = VPre + VSync + VPost + VDisplay + VBlank = 14 + 4 + 4 + 1200 + 38 = 1260 = even

5 Backporch > (TransferDelay + 1)

TransferDelay is implementation dependent, and has not been calculated here.

6 For IMS G332/364:

BackPorch > (TransferDelay + 1)

for IMS G335:

BackPorch > (TransferDelay + 2)

TransferDelay is implementation dependent, and has not been calculated here.

7 With the G332 and G364, Backporch > 16.

7.3.4 SUMMARY OF G3xx PARAMETERS FOR 800×600 AT 35MHz

Register	G332/G364	G335
HalfSync	8	8
BackPorch	31	31
Display	200	200
Linetime	250	250
ShortDisplay	77	77
BroadPulse	108	108
VPre	14	14
VSync	4	4
VPost	4	4
VBlank	38	38
VDisplay	1200	1200
(FrontPorch)	3	3

7.3.5 PARAMETER CALCULATIONS FOR 800×600 AT 36MHz

Horizontal parameters **800×600 at 36MHz**

1 period notSerialClock (1 Screen Unit) = 4 / 36MHz = 111.11ns.

Display = horizontal resolution / 4 = 800 / 4 = 200.

This gives a horizontal display period of 200 × notSerialClock = 22.22μs.

Linetime = A / notSerialClock = 28.44μs / 111.11ns = 255.97

Rounding to nearest even integer, Linetime = 256. This gives a horizontal period of Linetime × notSerialClock = 256×111.11ns = 28.44μs (which corresponds to 35.16KHz). The period 28.44μs can therefore be used for A in subsequent horizontal calculations.

HalfSync = (B / notSerialClock) / 2 = (2μs / 111.11ns) / 2 = 9.

This gives a horizontal sync pulse width of (HalfSync × notSerialClock) × 2, hence horizontal sync = 2μs; exactly the specified value.

BackPorch = C / notSerialClock = 3.56μs /ns = 32.04.

Rounding to nearest integer BackPorch = 32. This gives a horizontal back porch of BackPorch × notSerialClock = 3.56μs.

continued

Horizontal parameters (continued) **800×600 at 36MHz**

FrontPorch does not have to be explicitly set in the IMS G3xx products, but will be automatically set to:

FrontPorch = Linetime – ((2 × HalfSync) + BackPorch + Display)

FrontPorch = 256 – ((2 × 9) + 32 + 200) = 6.

This gives a horizontal front porch period of FrontPorch×notSerialClock = 6×111.11ns = 0.67µs.

ShortDisplay < (Linetime / 2) – ((2×HalfSync) + BackPorch)

ShortDisplay < (256 / 2) – ((2 × 9) + 32) < 128 – 50 < 78
Therefore ShortDisplay = 77

This is not specified in the specification, but must be programmed into the G3xx.

BroadPulse = (Linetime / 2) – (2 × HalfSync)

BroadPulse = (256 / 2) – (2×9) = 128 – 18 = 110

This is not specified in the specification, but must be programmed into the G3xx.

Horizontal frequency

Using these parameters, the horizontal period and thus frequency will be:

Horizontal frequency = 1 / horizontal period

h period = Linetime × notSerialClock

h period = 256 × 111.11ns = 28.44µs

Horizontal frequency = 35.156KHz

Vertical parameters **800×600 at 36MHz**

One period HalfLinetime = 28.44µs / 2 = 14.22µs.

Note that the period HalfLinetime is not a parameter or register set in the G3xx devices, but a period used in the calculation of the vertical parameters.

R = vertical resolution × Linetime × notSerialClock

R = 600 × 256 × 111.11ns = 17.07ms.

Setting the period S to be (1 / 56) – (P + Q + R):

S = (1 / 56) – (0.06 + 0.6 + 17.07)ms = 0.127ms.

VSync = P / HalfLinetime = 0.06ms / 14.22µs = 4.219

Rounding to nearest integer, VSync = 4. This gives a vertical sync pulse width of VSync × HalfLinetime = 4 × 14.22µs = 0.057ms.

continued

Vertical parameters (continued) **800×600 at 36MHz**

VPre and VPost

VPre = S / HalfLinetime = 0.127ms / 14.22μs = 8.94

Rounding to the nearest even integer, VPre = 8. This gives a VPre period of 0.114ms which compares to the (revised) specification of 0.127ms.

VPost can be set to the same as VSync, as it is not explicitly defined in the standard, so VPost = 4. This gives a VPost period of 0.057ms.

VDisplay = vertical resolution × 2 (in halflines) so VDisplay = 1200

Rounding to the nearest integer, VDisplay = 1200. This gives a vertical display period = VDisplay × HalfLinetime = 1200 × 14.22μs = 17.064ms.

VBlank = (O / HalfLinetime) – (VSync + VPre + VPost + VDisplay)

VBlank = (17.86ms / 14.22μs) – (4 + 8 + 4 + 1200) = 39.78

Rounding this to the nearest even integer, VBlank = 40. This gives a vertical blanking time of VBlank × HalfLinetime = 40 × 14.22μs = 0.569ms.

Vertical frequency

The vertical frequency obtained using these vertical parameters is thus:

Vertical frequency = 1 / vertical period

Vertical period = (VSync+VPre+VPost+VDisplay+VBlank) × HalfLinetime

Vertical period = 1256 × 14.22μs = 17.863ms.

Vertical frequency = 1 / 17.863μs = 55.98Hz

7.3.6 PARAMETER CHECKS FOR 800×600 AT 36MHz

1 All parameters are non–zero.

2 Linetime must be even.

3 Half line point:

Ha > (Linetime / 2) > Hb

where:

Ha = (2 × HalfSync) + Backporch + Display

Hb = (2 × HalfSync) + BackPorch

So:

Ha = (2 × 9) + 32 + 200 = 250

Hb = (2 × 9) + 32 = 50

So 250 > 128 > 50 so this condition is met.

4 Total of vertical lines per frame must be integer, i.e. number of half lines must be even:

Number of halflines per frame = VPre + VSync + VPost + VDisplay + VBlank = 2 + 4 + 4 + 1200 + 40 = 1250 = even

5 For IMS G332/364:

BackPorch > (TransferDelay + 1)

for IMS G335:

BackPorch > (TransferDelay + 2)

TransferDelay is implementation dependent, and has not been calculated here.

6 TransferDelay ≤ Short Display

TransferDelay is implementation dependent, and has not been calculated here.

7 With the G332 and G364, Backporch > 16.

7.3.7 SUMMARY OF G3xx PARAMETERS FOR 800×600 AT 36MHz

Register	G332/G364	G335	Register	G332/G364	G335
HalfSync	9	9	VPre	8	8
BackPorch	32	32	VSync	4	4
Display	200	200	VPost	4	4
Linetime	256	256	VBlank	40	40
ShortDisplay	77	77	VDisplay	1200	1200
BroadPulse	110	110	(FrontPorch)	6	6

7.4 CALCULATIONS FOR 1024×768 (SVGA)

7.4.1 TIMING SPECIFICATION

Horizontal						Vertical					
		Units			Units			Units			Units
Frequency	48.37	KHz	C	2.957	μs	Frequency	60	Hz	Q	0.54	ms
A	20.675	μs	D	15.95	μs	O	16.52	ms	R	15.88	ms
B	1	μs	E	0.75	μs	P	0.08	ms	S	0.02	ms
Misc: Composite sync on green, non-interlaced.											

Note that 1 / 60Hz = 16.67ms, not 16.52ms as in the specification. Therefore 16.67ms will be used for O. The period S will also be set to be (1 / 60) – (P + Q + R).

Preliminary calculations **1024×768 at 65MHz (SVGA)**

Given that:

- horizontal display = 15.95μs
- horizontal resolution = 1024
- video frequency = 1024 / 15.95μs = 64.2MHz

A video frequency of 65MHz will be assumed. Using a slightly higher video frequency means that the horizontal display period will be slightly less wide than if 64MHz had been used. The difference will be of the order of 1.5% which can easily be compensated for using the monitor display width adjustment. FrontPorch will be incremented to compensate for this.

Horizontal parameters **1024×768 at 65MHz (SVGA)**

1 period notSerialClock (1 Screen Unit) = 4 / 65MHz = 61.54ns

Display = horizontal resolution / 4 = 1024 / 4 = 256

This gives a horizontal display period of 256×notSerialClock = 15.75μs, which replaces the 15.95μs specified for subsequent calculations.

Using a horizontal frequency of 48.5KHz instead of the 48.37KHz specified, A = 20.62μs. This will be used in subsequent calculations.

continued

Horizontal parameters (continued) **1024×768 at 65MHz (SVGA)**

Linetime = A / notSerialClock = 20.62µs / 61.54ns = 335.05

Rounding to nearest even integer, Linetime = 336. This gives a horizontal period of Linetime × notSerialClock = 336 × 61.54ns = 20.68µs (48.36KHz), which compares to 20.675µs (48.37KHz).

HalfSync = (B / notSerialClock) / 2 = (1µs / 61.54ns) / 2 = 8.125

Rounding to nearest integer, HalfSync = 8. This gives a horizontal sync pulse width of (HalfSync × notSerialClock) × 2, hence horizontal sync = 0.98µs.

BackPorch = C / notSerialClock = 2.957µs / 61.54ns = 48.05

Rounding to nearest integer BackPorch = 48. This gives a horizontal back porch of BackPorch × notSerialClock = 2.954µs.

FrontPorch does not have to be explicitly set in the IMS G3xx products, but will be automatically set to:

FrontPorch = Linetime – ((2 × HalfSync) + BackPorch + Display)

FrontPorch = 336 – ((2 × 8) + 48 + 256) = 16.

This gives a horizontal front porch period of FrontPorch × notSerialClock = 16 × 61.54ns = 0.985µs, which exceeds 0.75µs, although not by a large amount (31%). The main reason for this discrepancy, as mentioned above, is the higher video data rate being used. Other reasons involve the rounding directions chosen in the above calculations (for HalfSync, BackPorch and Linetime).

ShortDisplay < (Linetime / 2) – ((2×HalfSync) + BackPorch)

ShortDisplay < (256 / 2) – ((2×8) + 48) < 128 – 64 < 64
Therefore ShortDisplay = 63

This is not specified in the specification, but must be programmed into the G3xx.

BroadPulse = (Linetime / 2) – (2×HalfSync)

BroadPulse = (256 / 2) – (2×8) = 128 – 16 = 112

This is not specified in the specification, but must be programmed into the G3xx.

Horizontal frequency

Using these parameters, the horizontal period and thus frequency will be:

Horizontal period = Linetime×notSerialClock

Horizontal period = 336×61.54ns = 20.68µs

Horizontal frequency = 1 / horizontal period = 48.36KHz

Vertical parameters **1024×768 at 65MHz (SVGA)**

One period HalfLinetime = 20.68µs / 2 = 10.34µs.

Note that the period HalfLinetime is not a parameter or register set in the G3xx devices, but a period used in the calculation of the vertical parameters.

R = vertical resolution × Linetime × notSerialClock

R = 768 × 336 × 61.54ns = 15.88ms (which equals the specification)

Setting the period S to be (1 / 60) – (P + Q + R):

S = (1 / 60) – (0.08 + 0.54 + 15.88)ms = 0.167ms

VSync = P / HalfLinetime = 0.08ms / 10.34µs = 7.74

Rounding to nearest integer, VSync = 8. This gives a vertical sync pulse width of VSync × HalfLinetime = 8 × 10.34µs = 0.083ms, which compares to 0.08ms.

VPre and VPost

VPre = S / HalfLinetime = 0.167ms / 10.34µs = 16.15

Rounding to the nearest integer, VPre = 16. This gives a VPre period of 0.165ms.

VPost can be set to the same as VSync, as it is not explicitly defined in the standard, so VPost = 8. This gives a VPost period of 0.083ms.

VDisplay = vertical resolution × 2 (in halflines) so VDisplay = 1536

This gives a vertical display period = VDisplay × HalfLinetime = 1536 × 10.34µs = 15.88ms.

VBlank = (O / HalfLineTIme) – (VSync + VPre + VPost + VDisplay)

VBlank = (16.67ms / 10.34µs) – (8 + 16 + 8 + 1536) = 44.1

Rounding this to the nearest even integer, VBlank = 44. This gives a vertical blanking time of VBlank × HalfLinetime = 44 × 10.34µs = 0.455ms.

Vertical frequency

The vertical frequency obtained using these vertical parameters is thus:

Vertical frequency = 1 / vertical period

Vertical period = (VSync+VPre+VPost+VDisplay+VBlank) × HalfLinetime

Vertical period = 1612 × 10.34µs = 16.67ms

Vertical frequency = 1 / 16.67ms = 60Hz

7.4.2 PARAMETER CHECKS FOR 1024×768 AT 65MHz

1 All parameters are non-zero.

2 Linetime must be even.

3 Half line point: Ha > (Linetime / 2) > Hb

where:

Ha = (2×HalfSync) + Backporch + Display

Hb = (2×HalfSync) + BackPorch

So:

Ha = (2×8) + 48 + 256 = 320

Hb = (2×8) + 48 = 64

So 320 > 128 > 64 so this condition is met.

Total of vertical lines per frame must be integer, i.e. number of half lines must be even:

Number of halflines per frame = VPre + VSync + VPost + VDisplay + VBlank = 2 + 8 + 8 + 1536 + 44 = 1598 = even

4 For IMS G332/364:

BackPorch > (TransferDelay + 1)

for IMS G335:

BackPorch > (TransferDelay + 2)

TransferDelay is implementation dependent, and has not been calculated here.

5 TransferDelay ≤ Short Display

TransferDelay is implementation dependent, and has not been calculated here.

6 With the G332 and G364, Backporch > 16.

7.4.3 SUMMARY OF G3xx PARAMETERS FOR 1024×768 AT 65MHz

Register	G332/G364	G335	Register	G332/G364	G335
HalfSync	8	8	VPre	16	16
BackPorch	48	48	VSync	8	8
Display	256	256	VPost	8	8
Linetime	336	336	VBlank	44	44
ShortDisplay	63	63	VDisplay	1536	1536
BroadPulse	112	112	(FrontPorch)	16	16

7.5 CALCULATIONS FOR 1280×1024

7.5.1 TIMING SPECIFICATION

Horizontal			Vertical			Horizontal			Vertical		
		Units			Units			Units			Units
Frequency	64	KHz	Frequency	60	Hz	C	2	μs	Q	0.57	ms
A	15.625	μs	O	16.67	ms	D	11.875	μs	R	16	ms
B	1	μs	P	0.08	ms	E	0.75	μs	S	0.02	ms

Preliminary calculations **1280×1024 at 105MHz**

Given that:

- horizontal display = 11.875μs
- horizontal resolution = 1280
- video frequency = 1280 / 11.875μs = 107.78MHz

A video frequency of 105MHz will be assumed. Using a slightly lower video frequency means that the horizontal display period will be slightly wider than if 107.78MHz had been used. The difference will be of the order of 2.6% which can easily be compensated for using the monitor display width adjustment.

Horizontal parameters **1280×1024 at 105MHz**

1 period notSerialClock (1 Screen Unit) = 4 / 105MHz = 38.1ns

Display = horizontal resolution / 4 = 1280 / 4 = 320.

This gives a horizontal display period of 320 × notSerialClock = 12.19μs, which exceeds the specified value of 11.875μs. This results from using 105MHz instead of 107.78MHz. The period 12.19μs must therefore be used for D in subsequent horizontal calculations.

Linetime = A / notSerialClock = 15.625μs / 38.1ns = 410.16

Rounding to nearest even integer, Linetime = 410. This gives a horizontal period of Linetime × notSerialClock = 410 × 38.1ns = 15.62μs (which corresponds to 64.02KHz). The period 15.62μs must therefore be used for A in subsequent horizontal calculations.

HalfSync = (B / notSerialClock) / 2 = (1μs / 38.1ns) / 2 = 13.125

Rounding to the nearest integer, HalfSync = 13. This gives a horizontal sync pulse width of (HalfSync × notSerialClock) × 2, hence horizontal sync = 0.99μs.

BackPorch = C / notSerialClock = 2μs / 38.1ns = 52.5

Rounding to nearest integer BackPorch = 52. This gives a horizontal back porch of BackPorch × notSerialClock = 1.98μs.

FrontPorch does not have to be explicitly set in the IMS G3xx products, but will be automatically set to:

FrontPorch = Linetime – ((2 × HalfSync) + BackPorch + Display)

FrontPorch = 410 – ((2 × 13) + 52 + 320) = 12

This gives a horizontal front porch period of FrontPorch × notSerialClock = 12 × 38.1ns = 0.457μs, which is lower than 0.75μs, although not by a large amount. The main reason for this discrepancy is the lower video data rate being used.

ShortDisplay < (Linetime / 2) – ((2×HalfSync) + BackPorch)

ShortDisplay < (410 / 2) – ((2×13) + 52) < 205– 78 < 127

Therefore ShortDisplay = 126

This is not specified in the specification, but must be programmed into the G3xx.

BroadPulse = (Linetime / 2) – (2×HalfSync)

BroadPulse = (410 / 2) – (2×13) = 210 – 26 = 184

This is not specified in the specification, but must be programmed into the G3xx.

continued

Horizontal parameters (continued) **1280×1024 at 105MHz**

Horizontal frequency

Using these parameters, the horizontal period and thus frequency will be:

Horizontal period = Linetime×notSerialClock

Horizontal period = 410 × 38.1ns = 15.62μs

Horizontal frequency = 1 / horizontal period = 64.02KHz

Vertical parameters **1280×1024 at 105MHz**

One period HalfLinetime = 15.62μs / 2 = 7.81μs

Note that the period HalfLinetime is not a parameter or register set in the G3xx devices, but a period used in the calculation of the vertical parameters.

R = vertical resolution×Linetime×notSerialClock

R = 1024×410×38.1ns = 16ms (which equals the specification)

Setting the period S to be (1 / 60) – (P + Q + R):

S = (1 / 60) – (0.08 + 0.58 + 16)ms = 0.007ms.

This is far below the specified value (0.02ms) (because of the lower video rate being used), so the value for VPost (period Q) can be reduced from 0.57ms to 0.56ms which allows S to be increased from 0.007ms to 0.017ms, which is closer to the specified value of 0.02ms.

VSync = P / HalfLinetime = 0.08ms / 7.81μs = 10.24

Rounding to nearest integer, VSync = 10. This gives a vertical sync pulse width of VSync × HalfLinetime = 10 × 7.81μs = 0.0781ms, which compares to the specification of 0.08ms.

VPre and VPost

VPre = S / HalfLinetime = 0.017ms / 7.81μs = 2.18

Rounding to the nearest (even) integer, VPre = 2. This gives a VPre period of 0.0156ms which compares to the specification of 0.02ms.

VPost can be set to the same as VSync, as it is not explicitly defined in the standard, so VPost = 10. This gives a VPost period of 0.0781ms.

VDisplay = vertical resolution × 2 (in halflines) so VDisplay = 2048

This gives a vertical display period = VDisplay × HalfLinetime = 2048 × 7.81μs = 15.99ms, which compares to the standard of 16ms.

continued

Vertical parameters (continued) **1280×1024 at 105MHz**

VBlank = (O / HalfLineTIme) – (VSync + VPre + VPost + VDisplay)

VBlank = (16.67ms / 10.34µs) – (8 + 16 + 8 + 1536) = 44.1

Rounding this to the nearest even integer, VBlank = 44. This gives a vertical blanking time of VBlank × HalfLinetime = 44 × 10.34µs = 0.455ms.

Vertical frequency

The vertical frequency obtained using these vertical parameters is thus:

Vertical frequency = 1 / vertical period

Vertical period = (VSync+VPre+VPost+VDisplay+VBlank) × HalfLinetime

Vertical period = 1612 × 10.34µs = 16.67ms

Vertical frequency = 1 / 16.67ms = 60Hz

7.5.2 PARAMETER CHECKS FOR 1280×1024 AT 105MHz

1 All parameters are non-zero.

2 Linetime must be even.

3 Half line point: Ha > (Linetime / 2) > Hb

where:

Ha = (2×HalfSync) + Backporch + Display

Hb = (2×HalfSync) + BackPorch

So:

Ha = (2×13) + 52 + 410 = 488

Hb = (2×13) + 52 = 78

So 488 > 205 > 78 so this condition is met.

4 Total of vertical lines per frame must be integer, i.e. number of half lines must be even:

Number of halflines per frame = VPre + VSync + VPost + VDisplay + VBlank = 2 + 10 + 10 + 2048 + 64 = 2134 = even

5 For IMS G332/364:

BackPorch > (TransferDelay + 1)

for IMS G335:

BackPorch > (TransferDelay + 2)

TransferDelay is implementation dependent, and has not been calculated here.

6 TransferDelay ≤ Short Display

TransferDelay is implementation dependent, and has not been calculated here.

7 With the G332 and G364, Backporch > 16.

7.5.3 SUMMARY OF G3xx PARAMETERS FOR 1280×1024 AT 105MHz

Register	G332/G364	G335
HalfSync	13	13
BackPorch	52	52
Display	320	320
Linetime	410	410
ShortDisplay	126	126
BroadPulse	184	184
VPre	2	2
VSync	10	10
VPost	10	10
VBlank	64	64
VDisplay	2048	2048
(FrontPorch)	12	12

7.6 CONCLUSIONS

The IMS G3xx family of devices can easily drive PC standard displays and, as shown in this Application Note, the calculation of the display parameters is straightforward.

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

DRIVING PAL AND NTSC WITH CVCS

GRAPHICS APPLICATIONS GROUP, BRISTOL

8.1 INTRODUCTION

This Application Note applies to the following SGS-THOMSON Color Video Controllers (CVCs) and describes how they may be configured to drive PAL [2] and NTSC video display devices:

- **IMS G332**
- **IMS G364**
- **IMS G335**

Throughout this document, information that is generic to all devices will be referred to by the collective name IMS G3xx. Where specific information pertains only to one more particular devices, the actual name of the device(s) will be specified.

This document should be read in conjunction with the relevant Datasheet for the particular device required [1].

8.2 IMS G3xx COLOR VIDEO CONTROLLERS

IMS G3xx CVCs are dedicated support chips that provide all necessary functions for controlling the real time raster operation of raster scan video systems using dual ported video DRAMs (VRAM). The facilities provided are designed to isolate the host processor from the constraints of the real time system without in any way interfering with ability of the processor to specify and manipulate screen data.

The devices provide the following functions:

- Programmable Video Timing Generator (VTG)
- Triple 256 location 8-bit lookup table (LUT)
- Triple 8-bit video DAC with anti-sparkle
- Phase-Locked Loop (PLL)
- Hardware cursor
- Video data path test

In terms of VRAM support, IMS G3xx CVCs allow for VRAM increment values (in interlaced mode) to be 1, 2, 256, 512 and 1024. This enables a wide variety of VRAM devices to be used, as well as providing for 2 and 4 Mbit VRAM upgrades. Also, this means that the logical screen width can be set to the display screen width (providing it is a power of 2) thus enabling software to map a 2 dimensional array directly onto the viewable portion of the VRAM display memory.

8.3 CONFIGURING FOR INTERLACE MODE

In order to configure IMS G3xx CVCs in interlace mode, the following register bits and parameters should be modified.

8.3.1 CONTROL REGISTER A

Interlace must be enabled by using the Control Register A Screen format bit (bit 1). This should be set to 1 to enable interlace mode.

The interlace standard may be defined by using the Interlace standard bit (Control Register A, bit 2). Setting to 0 defines EIA format setting to 1 defines CCIR format.

The appropriate VRAM address step increment must be selected using the VRAM address increment bits (Control Register A, bits [13:12]). The precise value of these bits is dependent upon the hardware design of the graphics subsystem and must therefore be provided by the hardware designer.

8.3.2 MEMINIT AND TRANSFERDELAY

MemInit and TransferDelay normally (in non-interlace mode) sum to the VRAM shift register length (expressed in multiples of 4 pixels) for a packed VRAM organization and mid-line updates. In interlace mode, however, they must sum to the value of Display (which is the displayed width, again in multiples of 4 pixels). This means that TransferDelay is set to the normal value calculated, and MemInit set to Display – TransferDelay (see [4]).

8.3.3 VSYNC, VBLANK AND VDISPLAY

VDisplay is normally (in non-interlace mode) set to twice the required display height (as this register value is in Half lines). With interlace mode, VDisplay must be set to the number of displayed Half lines in one field, and must be odd.

The sum of the VSync, VPre, VPost, VBlank and VDisplay periods must be odd. This means that as VSync, VPre and VPost values are in Half lines:

$$sum = \frac{VSync + VPre + VPost + VBlank}{2} + VDisplay$$

All other parameters are calculated as normal.

8.4 VIDEO FREQUENCY

Using interlacing enables the use of lower video frequencies than could otherwise be used using non-interlace modes.

IMS G3xx CVCs can easily be run at low video frequencies (less than 10MHz). In such cases the precise frequency should preferably be obtained by using an input clock frequency that corresponds to the full dot-rate and disabling the PLL (bit 5 = 0 in the Boot location register). Alternatively a low frequency clock may be used (e.g. 2.25MHz) and be multiplied up to the desired dot-rate by the PLL (Boot location register bit 5 = 1). Bits 0–4 specify the Binary coded PLL multiplication factor (subject to the input PLL clock range).

8.5 ELECTRICAL ASPECTS OF DRIVING NTSC AND PAL SYSTEMS

IMS G3xx CVCs drive R, G, and B color outputs. With systems requiring a single composite color signal, these must be put through suitable circuitry to provide the required signal. The Motorola MC1377 or Sony CXA 1145 color television RGB to PAL / NTSC encoder device provides this function, and this can be designed into a small circuit to produce the composite signal [3].

Care must also be taken to ensure that when synchronizing to another PAL source, the levels of sync signal used are compatible. IMS G3xx CVCs drive syncs at TTL levels, whereas some PAL systems expect negative going sync levels.

IMS G3xx CVCs drive their sync outputs with signal transition times that are too fast for the PAL specification, which states approximately 200ns. These can be slowed by connecting a capacitor of 1 - 2nF from the signal to ground. The precise value of this capacitor is however dependent upon the particular monitor connected. The requirement is to ensure that the switching times be set to the correct value of approximately 200ns.

If pure video and separate composite sync is required, Control Register A bit 6 (Analog video format) should be set to logic 1. Each RGB output would then just drive the video, and composite sync would be driven on the **notCorHSync** pin. Otherwise composite sync and video is produced on each of the RGB outputs.

8.6 CONFIGURING TO DRIVE PAL DISPLAYS

The CVC screen description parameters to produce PAL compatible signals are calculated and discussed below, and a complete list of the parameters is given.

Table 8.1. The PAL specification

Aspect ratio, width to height	4:3	Horizontal blank	12.05μs
Active vertical resolution	575	Horizontal sync	4.7μs
Line frequency	15.625KHz (64μs)	Horizontal frontporch	1.65μs
Frame frequency	50Hz	Vertical sync	2.5 lines
Horizontal sync + backporch	10.4μs	Vertical blank	25 lines

8.6.1 PARAMETER CALCULATIONS

Preliminary calculations **PAL**

Active Horizontal resolution = (575 × 4) / 3
Rounding up to nearest integer multiple of 4, hence
Horizontal resolution = 768

Video frequency period = (line time – horizontal blanking) / horizontal resolution
= ((1 / 15.625 KHz) – 12.05μs)) / 768
= 67.64ns
Video frequency = 14.783MHz

Horizontal backporch = 10.4 – 4.7μs = 5.7μs

Line frequency PAL

The PAL specification states an extremely tight tolerance on the line frequency of ± 0.02%. This is calculated first.

Using a video frequency of 14.78MHz yields a notSerialClock period (1 Screen Unit) of 4 × video frequency period = 270.636ns.

LineTime = horizontal period / notSerialClock

= 64μs / 270.636ns
= 236.48

Rounding to the nearest even integer, hence
LineTime = 236

Consequently, the line frequency is:

Line frequency = 1 / (LineTime × notSerialClock)

= 1 / (236 × 270.636ns)
= 15656.8 Hz

This is 31.8Hz or 0.2% above the PAL specification of 15625Hz. The PAL specification for the line frequency is thus not satisfied with these parameters.

Working backwards, the video frequency period can be calculated as:

Video Frequency period = (1 / 15625) / (236 × 4)
= 67.797ns

rounding to the nearest decimal place,

Video Frequency period = 67.8ns, so
Video Frequency = 14.75MHz

hence

Line frequency = 1 / (LineTime × notSerialClock)

= 1 / (236 × 271.2ns)
= 15624.22Hz

This is now 0.78Hz or 0.005% below the PAL specification, and is thus well within the specified tolerance.

Calculated IMS G3xx register values

The following pages show the calculations for the IMS G3xx screen description parameters. Note that the horizontal parameters are in multiples of notSerialClock periods (Screen Units).

Horizontal parameters PAL

1 period notSerialClock (1 Screen Unit) = 4 / 14.75MHz = 271.2ns.

HalfSync = (Horizontal sync / 2) / notSerialClk

= (4.7µs / 2) / 271.2ns
= 8.666
rounding to the nearest integer,
HalfSync = 9

Check, (9 × 2) × 271.2ns = 4.88µs, which is within the specification.

BackPorch = (10.4µs – horizontal sync) / notSerialClk)

= 20.354
rounding to the nearest integer,
BackPorch = 20

Check, 20 × 271.2ns = 5.424µs, not actually specified in spec, but:
backporch + horizontal sync = 4.88 µs + 5.424µs
= 10.3 µs

Display = horizontal resolution / 4

= 768 / 4
= 192

LineTime = horizontal period / notSerialClock
=236
(For calculation see previous section 'Line frequency')

FrontPorch = LineTime – ((2 × HalfSync) + Display + BackPorch)

= 236 – ((2 × 9) + 192 + 20)
= 6

Check, 6 × 271.2ns = 1.627µs, within specification.

ShortDisplay < (LineTime / 2) – (2 × HalfSync + BackPorch + FrontPorch) –1

< 118 – (18 + 20 + 6) – 1
< 73
Therefore ShortDisplay = 72

BroadPulse = (LineTime / 2) – (2 × HalfSync)

= 118 – 18
= 100

Horizontal frequency = 1 / (LineTime × notSerialClock
=15624.22Hz
(For calculation see previous section 'Line frequency')

Vertical parameters **PAL**

VSync = $2 \times \text{VertSync}$

$= 2 \times 2.5$
$= 5$

VPre and VPost

VPre = VSync = 5

VPost = VSync = 5

VBlank = $(2 \times \text{VertBlank}) - (3 \times \text{VertSync})$

$= (2 \times 25) - (3 \times 2.5)$
$= 35$

VDisplay = $((\text{Total Lines} - \text{non displayed lines}) / 2) \times 2$

$= 625 - (\text{VBlank} + (3 \times \text{VertSync}))$
$= 625 - (35 + 15)$
$= 575$

8.6.2 SUMMARY OF IMS G3xx PARAMETERS

The parameters to drive PAL with IMS G3xx CVCs are shown in Table 8.2, together with the actual primary parameters being used.

Table 8.2. Summary of IMS G3xx parameters to drive PAL

Primary Parameters (PAL)		IMS G3xx Register values	
Height	575	HalfSync	9
Width	768	BackPorch	20
Frame rate	50Hz	Display	192
Line rate	15.624KHz	ShortDisplay	72
Video rate	14.75MHz	BroadPulse	100
Hor Sync	4.88μs	VPreEqualize	5
Hor Backporch	5.424μs	VSync	5
Vert Sync	2.5 lines	VPostEqualize	5
		VBlank	35
		VDisplay	575
		LineTime	236
		TransferDelay[1]	d
		MemInit	192 – d
		Control Register A[2]	xxx7

NOTES:

[1] Transfer delay is implementation dependent

[2] Bits 4 -23 of Control Register A should be set according to system requirements, full details are given in the relevant datasheet.

8.7 CONFIGURING TO DRIVE NTSC DISPLAYS

Calculating the IMS G3xx parameters to drive NTSC, at 640 (displayed width) by 525 (displayed height) resolution, at 12.17MHz is straightforward, given the NTSC specification shown in Table 8.3.

The parameters are calculated and discussed below, and a complete list of the parameters to produce NTSC compatible signals is given.

Table 8.3. The NTSC specification

Aspect ratio, width to height	4:3	Horizontal BackPorch	4.7µs
Line frequency	15.73KHz	Horizontal sync	4.7µs
Duration of digital active line	640 pixels	Vertical sync	3 lines
Frame frequency	60Hz	Vertical blank	25 lines
Active vertical resolution	525	Video rate	12.17MHz

8.7.1 PARAMETER CALCULATIONS

Line frequency **NTSC**

Using a video frequency of 12.17MHz yields a notSerialClock period of 4 × video frequency period = 328.677ns, and thus:

LineTime = horizontal period / notSerialClock

= 63.57µs / 328.677 ns
= 193.4

Rounding to the nearest even integer, hence
LineTime = 194

Consequently the line frequency is:

Line frequency = 1 / (LineTime × notSerialClock)

=1 / (194×328.677ns)
=15682.9Hz

Calculated IMS G3xx register values

The calculations for the IMS G3xx screen description parameters follow. Note that the horizontal parameters are in multiples of notSerialClock periods (Screen Units).

Horizontal parameters **NTSC**

1 period notSerialClock (1 Screen Unit) = 4 / 12.17MHz = 328.677ns.

HalfSync = (Horizontal sync / 2) / notSerialClk

HalfSync = (4.7μs / notSerialClk) / 2
= 7

BackPorch = 4.7μs / notSerialClk

= 14

Display = horizontal resolution / 4

= 640 / 4
= 160

LineTime = horizontal period / notSerialClock
= 194
(For calculation see previous section 'Line frequency')

FrontPorch = LineTime – ((2 × HalfSync) + Display + BackPorch)

= 194 – (14 + 160 + 14)
= 6

ShortDisplay < (LineTime / 2) – (2 × HalfSync + BackPorch + FrontPorch)–1

< 97– (14 + 14 + 6) –1
< 62
Therefore ShortDisplay = 61

BroadPulse = (LineTime / 2) – (2 × HalfSync)

= 97 – 14
= 83

Horizontal frequency = 1 / (LineTime × notSerialClock
=15682.9Hz
(For calculation see previous section 'Line frequency')

Vertical parameters **NTSC**

VSync = 2 × VertSync

= 2 × 3
= 6

VPre and VPost

VPre = VSync = 6

VPost = VSync = 6

VBlank = Vertical total – VPre – VPost – VSync – VDisplay

= 525 – 6 – 6 – 6 – 485
= 22

VDisplay = ((Total Lines – non displayed lines) / 2) × 2

= 525 – (VBlank + VertSync + VPre + VPost)
= 525 – 40
= 485

8.7.2 SUMMARY OF IMS G3xx PARAMETERS

Table 8.4.

Primary Parameters (NTSC)		**IMS G3xx Register values**	
Height	525	HalfSync	7
Width	640	BackPorch	14
Frame rate	60Hz	Display	160
Line rate	15.73KHz	ShortDisplay	61
Video rate	12.17MHz	BroadPulse	83
Hor Sync	4.7μs	VPreEqualize	6
Hor Backporch	4.7μs	VSync	6
Vert Sync	3 lines	VPostEqualize	6
		VBlank	22
		VDisplay	485
		LineTime	194
		TransferDelay[1]	d
		MemInit	160 – d
		Control Register A[2]	xxx3

NOTES:

[1] Transfer delay is implementation dependent

[2] Bits 4 -23 of Control Register A should be set according to system requirements, full details are given in the relevant datasheet.

These values yield a full screen picture. The value of horizontal backporch can be adjusted, if required, to horizontally centralize the picture.

8.8 CONCLUSION

This Application Note has described how interlacing mode may be enabled on IMS G3xx CVCs and also how to use this functionality to drive PAL and NTSC display devices. Note that other resolutions and video frequencies can be used with NTSC and PAL systems by recalculating the G3xx parameters appropriately.

8.8.1 PAL

All the timings specified in the PAL specification have been verified to be correct both mathematically and using an oscilloscope. These parameters yield a full screen picture, and have successfully driven several televisions and video cassette recorders tested. A PAL compatible camera has also been synchronized to, for video mixing purposes.

8.8.2 NTSC

All the timings specified in the NTSC specification have been verified to be correct both mathematically and using an oscilloscope. The CVC frame timing signal, **FrameInactive** was verified to be 60Hz. These parameters yield a full screen picture, and have successfully driven an NTSC compatible television.

8.9 REFERENCES

1 *The Computer Graphics Databook,* 3rd edition, SGS-THOMSON, November 1992

2 *Specification of Television Standards for 625-Line System 1 Transmissions in the United Kingdom*, Department of Trade and Industry, 1984.

3 *Motorola MC1377 color television RGB to PAL/ NTSC encoder*, Datasheet, Motorola Inc., 1986.

4 *Using the IMS G335 automatic screen refresh,* Application Note, SGS-THOMSON, February 1993 (Chapter 9 in this volume).

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

USING THE CVC AUTOMATIC SCREEN REFRESH

GRAPHICS APPLICATIONS GROUP, BRISTOL

9.1 INTRODUCTION

The IMS G335 Color Video Controller (CVC) provide all the necessary signals and support for VRAM based framestores. Programmable framestore description parameters are used in providing intelligent control of VRAM row transfers to maintain a constant supply of pixels to the pixel interface. This automatic refresh allows a packed pixel framestore to be implemented for many different screen formats, independent of VRAM SAM register length. Simplified system design and reduced bus control time can be achieved through the support of the G335 for split SAM VRAMs.

This document is intended to provide help in programming the IMS G335 for optimal operation and explain the transfer cycle operation of the Framestore Manager. Some familiarity of the G335 Framestore Manager registers is assumed. Full information on these is provided in the IMS G335 datasheet contained in the Computer Graphics Databook [2].

9.1.1 BASIC VRAM ROW TRANSFER OPERATION

VRAMs contain two portions of memory, DRAM (Dynamic Random Access Memory) and a SAM (Serial Access Memory) shift register. Data is loaded from the DRAM to the SAM in an special cycle called a DRAM to SAM transfer. This transfer cycle is usually initiated by holding the VRAM's DT/OE pin (or equivalent) low before RAS is active low. The address on the VRAM address pins at the falling edge of RAS dictate which row portion of the DRAM will be loaded into the SAM, whilst the address values on the VRAM address pins at the CAS falling edge specify the first data value from the SAMs on the next clock edge following the completion of the cycle.

Data is read serially from the SAMs by means of a ShiftClock (VRAM pin usually called SC), whose maximum frequency can be as high as 55MHz[1]. SAM lengths can be 256 or 512 bits long. A DRAM to SAM transfer operation is required when one of the following occurs:

1 Data from SAM is just about to be exhausted

2 Reload of SAM is required for a new video line

3 Special effects

In primitive video systems, it is not possible to perform SAM transfer updates during the active portion of the video display. These systems are therefore dependent on the length of the SAM and hence restrict the resolution of the screen regardless of the amount of video memory available. The G335 Framestore Manager imposes no such overhead; arbitrary screen resolutions can be displayed regardless of the SAM size, using a facility called midline updates. Flexible choice of screen resolutions is achieved by scheduling transfer cycles during active video display. Midline updates do not require split SAM VRAM operation, although this facility (found on most VRAMs) can be exploited by the G335.

9.1.2 G335 DMA TRANSFER OPERATION

Two registers on the G335, containing the framestore description parameters MemInit and TransferDelay, control the frequency and timing of SAM transfers. MemInit and TransferDelay represent time periods in Serial Clock (SClk) units, regardless of the bits per pixel or interleaving mode in which the G335 has been programmed. SClk always runs at one quarter of the pixel frequency (except in 24bpp mode on a 32-bit pixel port).

In non-split SAM mode the value of MemInit is the number of SClk periods required to expire before a transfer cycle is initiated whilst TransferDelay specifies the maximum possible time for a transfer cycle to complete. The period of transfer cycles is thus MemInit + TransferDelay SClk periods.

In split SAM mode MemInit specifies the period of transfer requests, whilst TransferDelay specifies the time to complete a transfer cycle.

When the G335 requires a VRAM transfer cycle it will assert **BusReq** indicating to any VRAM address bus master that the G335 requires access to the VRAMs. The G335 will only assert **BusReq** in the event of a VRAM transfer cycle being required.

The master should respond, within a known worse case time, by asserting **BusGranted/AOE**. This instructs the G335 that control of the VRAM address bus has now been relinquished and a relevant transfer address can be output on the **ADBus** lines of the G335.

Transfer strobe **DTA** (and if interleaving is active **DTB**) will go high some time after **BusGranted/AOE** is asserted. In non-split SAM mode these strobes control the exact moment when the transfer from DRAM to SAM will occur, which is critical in high speed non-split SAM systems. On the falling edge of these strobes, a transfer cycle is completed.

[1] Specified ShiftClock period of MICRON MT42C8256 is 18ns

In a split SAM system, the operation of a transfer cycle is similar to that described above except that the timing of the transfer cycle is not critical. As such **DTA** (or **DTB**) should not be used for the generation of transfer cycle strobes. Note that in a contiguous (packed pixel) framestore a single non-split SAM transfer cycle is requested at the start of a frame in the inactive portion of the video display. This ensures that all SAMs are primed with the correct video data at the start of the display.

9.2 PROGRAMMING THE G335 FRAMESTORE DESCRIPTION REGISTERS

As mentioned above, the two registers which control the frequency of transfer cycles (MemInit and TransferDelay) are programmed in units of SClk periods. This period is dependent on the pixel frequency that has been programmed into the G335.

9.2.1 TRANSFERDELAY VALUE

The value of TransferDelay is the maximum time required in SClk units to complete a transfer cycle; this is the sum of the system latency in responding to a **BusReq** signal and the time required to complete a VRAM transfer cycle. This time is generally fixed for all modes, but because of changes in SClk frequency for different screen resolutions, it must be recalculated in units of SClks such that TransferDelay must not be less than system latency plus transfer cycle time.

For example, in a typical graphics system:

$$\textbf{TransferDelay} = \frac{T_{lat} + (T_{VTC} \times P_{SClk})}{P_{SClk}} \quad \textbf{periods of SClk}$$

Where:
T_{lat} = System latency in responding BusGranted to BusReq (max) (ns)
T_{VTC} = Time to complete VRAM transfer cycle (SClks)
P_{SClk} = Period of SClk (ns)

For example:

T_{lat} = 900ns (say)

T_{VTC} = 4 SClks (say)

Video pixel frequency (Fp) = 110MHz (say)

Therefore Serial clock (SClk) frequency = Fp/4 = 27.5MHz

So P_{SClk} = 10^6/27.5 = 36.4ns

Therefore P_{SClk} = 36.4ns

$$\textbf{TransferDelay} = \frac{900\text{ns} + (4 \times 36.4 \times 10^{-9})}{36.4 \times 10^{-9}} \quad \textbf{periods of SClk}$$

TransferDelay = 28.73 SClks

TransferDelay (to nearest largest integer SClk) = 29 SClks

However, assume that the transfer cycle is a fixed value, not dependent on Serial Clock, for example:

T_{VTC} = 100ns (say)

then the TransferDelay value would be different:

TransferDelay (in ns) = T_{lat} + T_{VTC} = 900ns + 100ns = 1000ns

$$\textbf{TransferDelay} = \frac{(900 + 100) \times 10^{-9}}{36.4 \times 10^{-9}} \quad \textbf{periods of SClk}$$

TransferDelay in SClk units = 27.47 SClks

TransferDelay to nearest SClk = 28 SClks

9.2.2 MEMINIT VALUE

The value of MemInit is determined by the following criteria:

1 Whether contiguous framestore is required

2 Interleaved or non-interleaved mode

3 Size of SAM in VRAMs

4 Usage of split SAM

5 Whether interlacing is active

Contiguous memory in a video display is defined as being a framestore which contains the same address increment between successive pixels in the x direction for all pixels and the same increment for the last pixel of a line to the first pixel of the next successive line.

In a scrolling or panning system, contiguous memory usage is not always possible and this will therefore have an effect on the value placed into the MemInit register. Depending on the system architecture, it may be possible to use a different VRAM increment in order to use the next single power of 2 screen stride, as discussed in Section 9.2.3. This is because the Framestore Manager must perform a VRAM reload for every line, and cannot start arbitrarily through the VRAM shift register.

Contiguous and non-contiguous memory systems are shown in Figures 9.1 and 9.2 respectively. In a non-contiguous system the displayed portion of memory can be moved within the confines of the logical screen. This is achieved by modifying the value placed in TopofScreen. In this case however a transfer cycle request is required for every displayed line.

Figure 9.1. Contiguous framestore

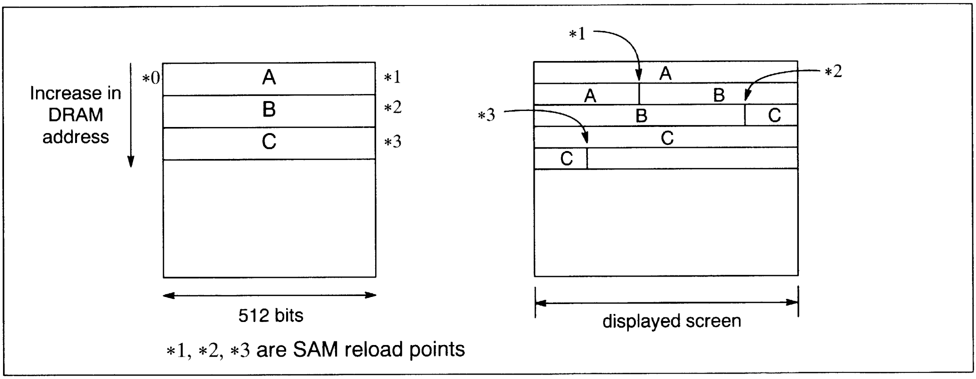
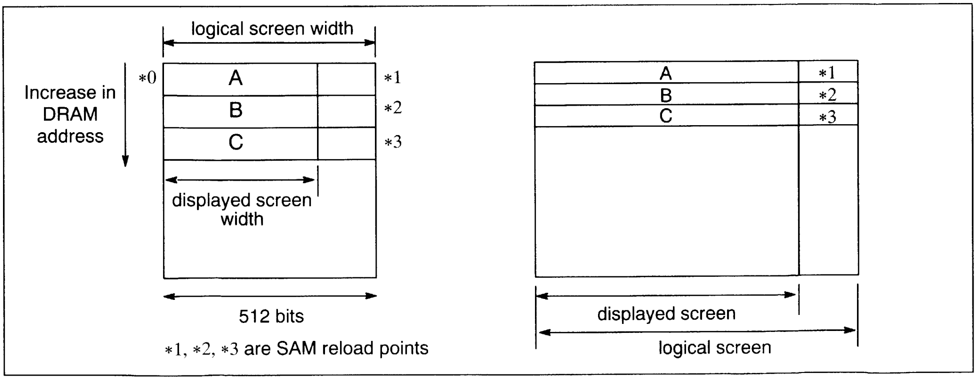


Figure 9.2. Non-contiguous framestore



Contiguous non-split SAM framestore usage

In a contiguous, non-split SAM framestore, the value of MemInit + TransferDelay specifies the period of transfer operations in order to update SAM pixel data. When fulfilling this requirement transfer cycles can occur at any time during active video. The method of deriving TransferDelay was described in Section 9.2.1. MemInit is dependent on SAM length; using Figure 9.3 and Table 9.1 as a reference and knowing the SAM length in the system, MemInit can be derived as shown in the example below.

Figure 9.3 shows the relationship of ShiftClock cycles to SClk cycles for a non-interleaved 32-bit wide pixel port system as on IMS G335, and Table 9.1 shows the relationship of MUX ratio to ShiftClock period.

Figure 9.3. notShiftClkA operation in non-interleaved pixel modes (32-bit wide pixel port)

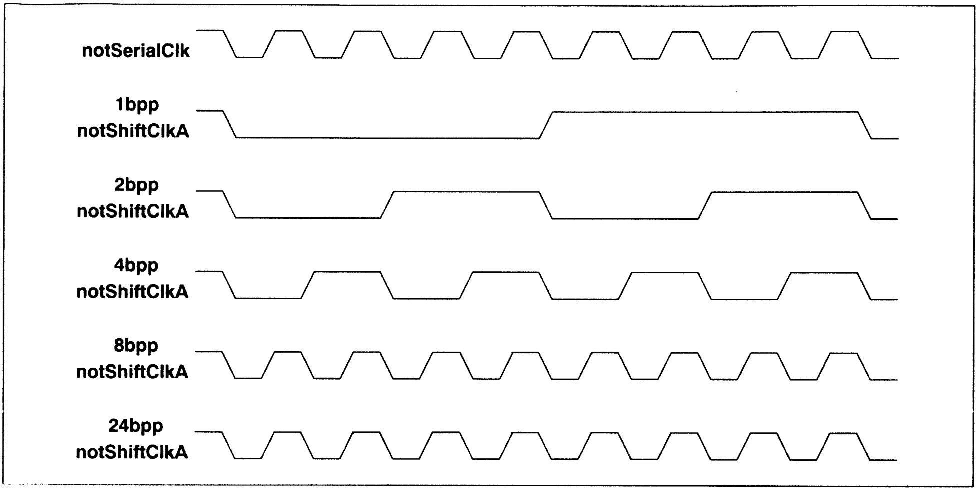


Table 9.1. Non-interleaved pixel modes (32-bit wide pixel port)

Control Register A bits				Bits per pixel	ShiftClk period	MUX ratio	Use of LUT
22	21	20	18				
1	1	0	0	24	1 SClk	1:1	Full color*
0	1	1	0	8	1 SClk	4:1	Pseudo color
0	1	0	0	4	2 SClks	8:1	Pseudo color
0	0	1	0	2	4 SClks	16:1	Pseudo color
0	0	0	0	1	8 SClks	32:1	Pseudo color
*Note that in this mode 1 Screen Unit (SU) = 1 pixel							

Suppose SAM length = 256 bits (for example)

TransferDelay = 32 SClks (say)

Bits per pixel = 8

MUX ratio = 4:1

Non-Interleaved, non-split SAM operation

MemInit = (SAM length × ShiftClk period) –
TransferDelay

From Table 9.1, ShiftClock period = 1 SClk

Therefore, period of transfer cycle = 256 × 1
= 256 SClks

Period of transfer cycle = MemInit + TransferDelay
= 256 SClks

MemInit = Period – TransferDelay = 256 – 32
= 224

So MemInit = 224

If the same system parameters are used but with an interleaved framestore, assuming the same video data rate (and hence SClk period), the size of the SAMs are effectively increased by two, i.e. from 256 bits to 512 bits. This means that the period of ShiftClock in units of SClk is also increased by a factor of two:

New value of (MemInit + TransferDelay) = 512

If TransferDelay remains = 32, then
New value of MemInit = 512 – TransferDelay

MemInit = 512 – 32

MemInit = 480

Contiguous split SAM framestore usage

Using the same system parameters as above but introducing split SAM operation, the SAM length is now halved before data is exhausted from one half of the SAM.

In this case MemInit (on its own) describes the period of split SAM transfer. TransferDelay must still be programmed with the correct (maximum delay + duration) value but is not used by the G335 in timing the period of transfers.

Using the above information it can be shown that:

MemInit = 256

(Control Register B bit 3 should be set to 1 to enable split SAM on the G335)

If an interleaved pixel port is not used, then:

MemInit = 128

9.2.3 NON-CONTIGUOUS FRAMESTORE

In a non-contiguous display, i.e. where panning is enabled or the display is interlaced, the logical screen width must be larger than the displayed screen width as shown in Figure 9.2. The logical screen width must be defined as using no more pixel information than the maximum amount of pixel information that is loaded into the SAMs during a transfer cycle. A transfer cycle must occur for every displayed line, so that information in the SAM is correctly refreshed with appropriate data for each displayed line.

This means that in a non-split SAM system MemInit + TransferDelay = Display, i.e. equals the duration of a displayed line. This means that for each displayed line a transfer cycle will occur in order to set SAM data at the correct position.

Panning is controlled by setting the TopOfScreen register to appropriate starting addresses. This then has the effect of moving the displayed portion of the screen over the logical screen; the displayed portion of the screen being in effect a viewing window on a larger screen.

In some systems using split-SAM in a panning environment is less straightforward. It must be ensured that the MemInit time expires while displaying from the first half of the SAM, otherwise the system will fail.

9.2.4 INTERLACED DISPLAYS

Interlacing is only a solution for systems which cannot produce the high bandwidths required to facilitate high resolution screens, or resolutions demanding high refresh rates.

The advantage of using interlacing is that the bandwidth required for a display is reduced by a factor of up to 2 while still retaining the apparent screen resolution. In this system odd numbered lines, making up an odd field, are interlaced with the even numbered lines of an even field. These two fields are refreshed alternately. Drawbacks of this system include its dependence on the persistence of the monitor's phosphor and the subsequent interpretation of the eye (horizontal lines, occurring predominantly in computer generated images, can be perceived to be strobing).

The G335 supports interlace mode in both TV standard and even field interlace formats. The logical screen format of an interlaced screen is very similar to that found in a panning system. Again a transfer cycle must be initiated for every displayed line but in this case the increment of addresses between lines is twice that of a non-interlaced system. Also an offset address is applied for the second field, as shown in Table 9.2.

Table 9.2. VRAM address increment as set by Control Register A, bits 12 and 13

Register Bit		Non-interlace	Interlace	
13	12	Increment	Increment	Second field offset
0	0	1	not used	
0	1	256	2	1
1	0	512	512	256
1	1	1024	1024	512

In a non-split SAM system MemInit + TransferDelay = Display. This forces a transfer cycle every displayed line, similar to a panning system, but with twice the address increment.

In panning and interlaced split SAM mode the following conditions apply:

1 Control Register A, bit 1 must be set to interlaced mode, Control Register A, bit 2 may be set to either EIA or CCIR standard and Control Register B, bit 3 must be set to split SAM operation.

2 The VRAM address increments are identical to those shown in Table 9.2, page 131 for non split SAM operation.

3 The VRAM should be configured to perform unsplit transfers (i.e. the VRAM DSF pin should be held low) even though split SAM is selected on the IMS G335. This enables the same **BusReq/ AOE** logic to be used for both non-interlaced and interlaced split SAM modes.

4 MemInit must be set to the same value as Display; this ensures that a row transfer occurs at the end of each line, enabling the Framestore Manager to generate the two interlaced display fields from a single contiguous framestore.

9.2.5 ADDRESS GENERATION

The G335 loads its internal address pointer with the value programmed into the TopOfScreen register before the start of each frame. This internal address is incremented with the VRAM address increment specified in bits 12 and 13 of Control Register A.

In split-SAM mode the address increment value automatically adjusts to half the specified amount when a split-SAM cycle is in progress, as shown in Table 9.3 (note that setting bits 12 and 13 = 00 is not allowed). For maximum system performance it is advisable to set the address increment value to be equal the VRAM SAM length.

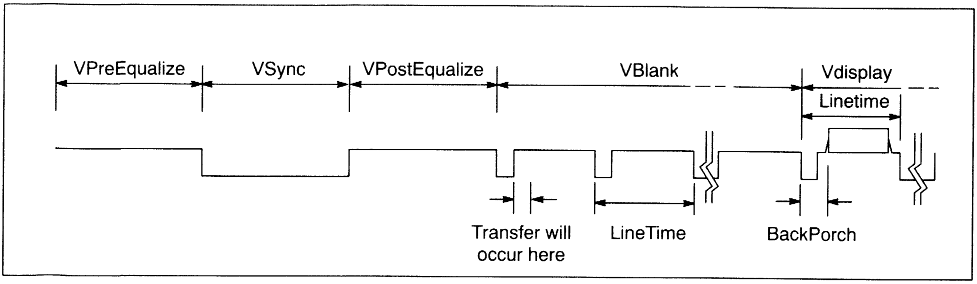
Table 9.3. Control of split SAM VRAM address increment

Register bit		Split SAM VRAM address increment	
13	12	First backporch in VBlank	During displayed screen
0	0	not allowed	not allowed
0	1	256	128
1	0	512	256
1	1	1024	512

9.2.6 G335 SCHEDULING OF SAM TRANSFER OPERATIONS

During the backporch of the VBlank period, a non-split SAM transfer cycle will initiate the SAMs for the first displayed line of a frame as shown in Figure 9.4. This will occur for all modes of SAM transfer.

Figure 9.4. Composite video waveform showing position of transfer cycle



In a synchronous transfer cycle system, when the MemInit time expires during active video, a transfer cycle will be initiated using the cycle duration specified in TransferDelay. The **DTA** (and **DTB**) transfer strobes will have an active edge 1/2 SClk cycle before the rising edge of **notShiftClockA** (and **B**). This procedure continues until the end of the visible display. The start of frame transfer cycle will then be activated as described earlier.

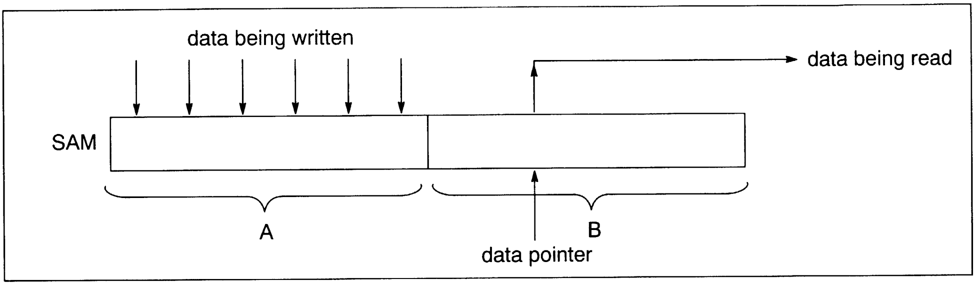
For split-SAM systems the operation is slightly different. At the start of a frame, a transfer cycle will occur as for non-split SAM systems, refreshing all portions of the SAM. During active video, MemInit time is monitored for expiry. When the MemInit time has expired a split SAM transfer cycle will be activated. Data will be transferred into the one half of the SAM while the other half is being used as a source of pixel data. In this case, as shown in Figure 9.5, SAM side B is being used as a source of pixel data, while data in side A which has just been read, will be refreshed with new data.

The address used to generate this first split SAM transfer in frame = TopOfScreen + (2 × split SAM Address Increment)

From this point the MemInit time will be monitored for expiry and a split SAM transfer will occur, but now using the specified split SAM address increment.

This procedure will continue until the end of the visible screen has been reached, following which a non-split SAM transfer cycle will occur during VBlank as described above.

Figure 9.5. Split SAM operation



9.3 CONCLUSION

This Application Note describes the programmable control of VRAM row transfer operations by the IMS G335 CVC. The G335 midline update facility enables a flexible choice of screen resolutions independent of VRAM SAM register length. This is achieved through the G335's ability to schedule transfer cycles during active video display.

The frequency and timing of VRAM SAM transfers is controlled by the G335 Framestore Manager using the programmable parameters MemInit and TransferDelay. This document describes how these parameters are determined for different system requirements including usage of contiguous framestore, interleaved or non-interleaved mode, size of SAM in VRAMs, usage of split-SAM and whether interlacing is active. Example calculations were used to illustrate the effect of these variables.

The generation of the VRAM transfer address by the G335 requires additional considerations in systems using panning, interlacing and split-SAM VRAM. The setting of the G335 VRAM address increment and TopofScreen register is described for these cases.

9.4 REFERENCES

1 *IMS G335 Preliminary information*, SGS-THOMSON, August 1992, Document number 42 1559 00

2 *Computer Graphics Databook*, 3rd edition, SGS-THOMSON, November 1992

Note that [1] is contained in the Computer Graphics Databook [2].

USING CVCs IN SLAVE MODE

GRAPHICS APPLICATIONS GROUP, BRISTOL

10.1 INTRODUCTION

This Application Note details how to use the IMS G332, G364 and G335 in slave mode. The essential principles are the same between the devices, as the slave locking mechanism is identical. In the following discussion, it may be helpful to refer to other related Application Notes within this notebook, as well as the datasheet for the appropriate device [1].

All the Color Video Controllers (CVCs) can operate in both master and slave modes, interlaced or non-interlaced, in all color modes and frequencies supported by that particular device.

Master mode is the mode normally used when the device is driving a graphics display, and is not required to synchronize to any other video source. Other video sources may themselves synchronize with this device, if required, using horizontal and vertical (or composite) TTL level sync signals output by the CVC.

Slave mode allows the CVC to synchronize to another video source, synchronizing on each frame, allowing the video sources to be manipulated, faded, and mixed with each other, and treated as one video source, perhaps being displayed on a single graphics display. Such uses might include using one video source to generate the main (background) display, and secondary sources to generate overlays on the main display, including the superimposing of text windows on a computer generated or live camera background picture. Land contours, direction finding information, and bomb sights could all be computer generated and displayed as a graphics overlay on a live image. If the two signals are not synchronized, then one will move in time with respect to the other (depending upon which is generating the sync signals being used), or will not appear at the correct place with respect to the other. On the graphics display, one video source will either not be seen at all, or will be seen rolling vertically and / or horizontally, and may be tearing, with respect to the other.

10.2 CVC MASTER MODE CONFIGURATION

The CVC is set to operate as a master by setting the *Device Operating Mode* bit (Control Register A bit 3) low (logic 0). This sets the composite and horizontal sync[1] (**notCorHSync**) and vertical sync (**notVSync**) pins to be outputs. Sync is generated internally, and these two sync pins are driven appropriately, at TTL levels, active low.

10.3 CVC SLAVE MODE CONFIGURATION

The CVC is set to operate as a slave by setting the *Device Operating Mode* bit (Control Register A bit 3) high (logic 1). This sets the composite and horizontal sync (**notCorHSync**) and vertical sync (**notVSync**) pins to be inputs.

10.3.1 NON-INTERLACED SLAVE MODE OPERATION

With non-interlaced slave mode operation, an external horizontal sync is not required, but an external vertical sync must be supplied, as the CVC synchronizes to each frame. The *Digital Sync Format* (controlled by Control Register A bit 5) can be either composite or separate, as required. The external vertical sync must be supplied to the **notVSync** input, and cannot be supplied to the **notCorHSync** pin.

10.3.2 INTERLACED SLAVE MODE OPERATION

When using interlaced slave mode operation, external horizontal *and* vertical sync signals must be supplied to the **notCorHSync** and **notVSync** pins respectively. The *Digital Sync Format* mode must be selected to be separate sync, by setting Control Register A bit 5 high (logic 1). The CVC will initially use the horizontal sync to determine which field is to be displayed, and subsequently synchronize on each field (not frame) using vertical sync.

10.4 SLAVE MODE SYNCHRONIZATION

10.4.1 EXTERNAL HSYNC PULSES DURING FRAME FLYBACK IN INTERLACED MODE

In interlaced slave mode on the IMS G332 or G364, external (master) HSync pulses should not be present during frame flyback or VSync. On the IMS G335 the *Flyback Waveform* mode (controlled by Control Register B bit 2) can be set (high) to allow HSync during flyback. This allows master HSync pulses during frame flyback.

If any pulses are present during frame flyback on the IMS G332 or G364, (or on the IMS G335 with *Flyback Waveform* set as no HSync during flyback) then correct field synchronization will not be possible. On the IMS G332 or G364, Master HSync pulses may be gated with the CVC's **FrameInactive** signal, perhaps using a simple OR gate, to

[1] This pin can be configured to be composite (both vertical and horizontal) sync, or horizontal sync, using the *Digital Sync Format* control bit (Control Register A bit 5).

eliminate any HSync pulses from occurring while **FrameInactive** is asserted. This problem does not arise when synchronizing multiple IMS G332 or G364 devices (or of course IMS G335 with *Flyback Waveform* set as no HSync during flyback) as the master CVC would not produce any HSync pulses during frame flyback.

Note that once the CVC has correctly synchronized to the master field order, then further HSync pulses are not actually required. VSync must however continue to be supplied. If the master HSync pulses are missing or out of the timing limits shown in Figures 10.1 and 10.2, then the VTG will always display the same single field. This is because the VTG's sync algorithm, having noted that it displayed the wrong field, will continue to display the same field as this is assumed to be correct.

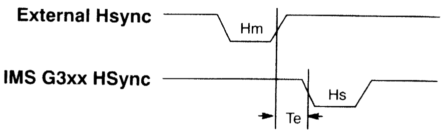
10.4.2 POSITION OF EXTERNAL HSYNC PULSES IN INTERLACED MODE

When in interlaced slave mode, the position of the external (master) HSync pulse with respect to the internally generated CVC HSync pulse at the DAC output[2] must be as shown in Figures 10.1 and 10.2. The external HSync pulse may occur anywhere between the earliest time and the latest time. (This period is partly comprised of the device programmable pipeline delay, *d*, as described below). The HSync pulse may have to be delayed using an analog delay line in order to meet the timing specification.

This timing specification must be observed for all HSync pulses present during each field. Ideally, just one HSync pulse should be allowed through to the CVC. This will allow the position of the single pulse to be adjusted to be exactly correct, and that no problems are encountered with drift or jitter during a field, causing a trailing pulse to move out of the timing specification. If only the *first* HSync pulse in the field is allowed through to the CVC, then this will be sufficient to correctly synchronize the slave field display. This can be achieved by triggering a flip-flop on the falling edge of **FrameInactive** which allows through one HSync pulse, and is then itself reset, disabling any further HSync pulses.

Early external HSync

Figure 10.1.

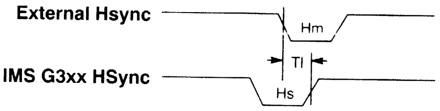


Where Te < (4.75 SClk + **notShiftClk** period + *d*)

The rising edge of the external HSync pulse must occur within the time Te before the falling edge of the internally generated CVC HSync pulse. The value for *d* is pixel pipeline delay of 0 -7 SClks[3] programmed using Control Register A bits [17:15].

Late external HSync

Figure 10.2.



Where Tl > (6 SClk + **notShiftClk** period + *d*)

The falling edge of the external HSync pulse must occur at time Tl before the rising edge of the internally generated CVC HSync pulse. The value for *d* is the pixel pipeline delay of 0 - 7 SClks programmed using Control Register A bits [17:15].

[2] The internally generated CVC HSync (and VSync) pulses can be seen on the DAC outputs if the *Analog Video Format* is set to video and sync composite.

[3] 1 SClk = 1 **notSerialClk** period = 4 pixels (except for 32-bit pixel port 24bpp mode).

Total time window for incoming HSync pulse

The total time window for the incoming (master) HSync pulse is thus:

```
Total window = Hm + Te + (Hs – Tl)
               (periods SClk)

For example assume that:
    Te = 5 periods SClk
    Tl = 7 periods SClk

then:
Total window = Hm + Hs – 2
               (periods SClk)
```

Where:

Hm is the width of the master HSync pulse

Hs is the width of the slave HSync pulse

Te is the earliest time

Tl is the latest time

What this means is that (subject to the video timing specifications being adhered to), the total time window for the incoming HSync pulse can be increased by increasing either (or both) of the master and slave HSync pulse widths.

10.4.3 PIXEL LEVEL SYNCHRONIZATION

When in slave mode, the CVC devices will lock onto an external active low, vertical sync signal. The synchronization is on each frame (or field in interlaced mode), not each line. Synchronization is guaranteed to be within one SClk period.

Synchronization to within one SClk period may not be adequate for the system being designed, in which case the setup and hold timing relationship between **notSerialClk** and the external syncs must be observed. See the appropriate data sheet for more details. (For the IMS G335 see Figure 13.37 on page 226 in the *Computer Graphics Databook, 3rd Edition*).

Alternatively the slave CVC may be clocked by a times one clock derived from the master video waveform, as discussed below.

10.4.4 VIDEO FREQUENCY CLOCKING

If different pixel clocks are used for the video sources (whether CVC devices or some other source), there will be the usual beating of one clock source against the other. With the CVC family of devices, this will result in either the master or slave display moving left and right horizontally, with respect to the other, depending upon which is generating the sync signals for the display. The movement will beat, which is when the different clock signals continually go in and out of phase. This is a common problem with distributed video systems using more than one video source that must be synchronized together.

The solution to this beating problem is to use a common video frequency clock, usually derived from the master video signal. This is discussed in greater detail in Section 10.6.

10.4.5 SLAVE SYNCHRONIZATION DELAY

When the CVC VTG starts, it will free run (produce syncs unrelated to the master syncs) until it receives a vertical sync signal, which will reset the VTG to the start of the vertical sync pulse. There will be a fixed delay between the vertical sync pulse being detected and the VTG restarting. This delay will be constant from then on. The delay (*n* SClks) is dependent on which device is being used and the bits per pixel, and will be between:

5 SClk + **notShiftClk** period + *d*
and
5.5 SClk + **notShiftClk** period + *d*

where *d* is the pixel pipeline delay of 0 - 7 SClks programmed using Control Register A bits [17:15].

Referring to the setup and hold timing relationship between **notSerialClk** and the external vertical sync signal as discussed in Section 10.4.3, the delay can increase by up to 1/2 SClk period, depending upon the exact timing of the external vertical sync signal being supplied.

The effect of the slave mode synchronization delay is to displace the slave display right by the appropriate number of pixels (*n* SClk periods), with respect to the master display. This effect may not be acceptable in certain applications, in which case it may be reduced by moving some or all of the delay time from the horizontal backporch to the frontporch, depending upon the video timing specifications that are being adhered to. If the slave is not producing any sync pulses, the whole delay value will *probably* be movable to the frontporch. With the CVC family of devices, this can be achieved by reducing the value programmed into BackPorch by *n*, keeping the resultant video backporch within specification, and the value for BackPorch > (Td + 2) (and FrontPorch ≥ 4 if cursor operation is enabled). The value of *n* SClks is discussed above. This will automatically increase FrontPorch by the time delta, this *not* having to be explicitly programmed into a CVC register.

10.5 SYNCHRONIZATION PULSES

10.5.1 SYNCHRONIZATION PULSES AND DAC OUTPUTS

Synchronization signals are generated in both master and slave modes. The **notCorHSync** and **notVSync** pins can be used to drive (or be driven by) separate horizontal and vertical sync respectively, or **notCorHSync** used to drive composite sync, depending upon the state of the *Digital Sync Format* (controlled by Control Register A bit 5). Note that if in master mode on the IMS G332 or G364, the **notVSync** output drives all the time, irrespective of the *Digital Sync Format*, and cannot be disabled. Note that the constant driving of **VSync** will only be a problem when driving certain monitors that do not display when driven with more than one (identical) sync signal (composite on video plus separate on sync pins). On the IMS G335 the sync outputs may be tristated (together) using Control Register B bit 9.

Composite sync can be superimposed onto all three DAC outputs, depending upon the state of the *Analog Video Format*, controlled by Control Register A bit 6. Note that on the IMS G335 sync may be selected to only be on the green DAC output, controlled by Control Register B bit 1. This composite sync is independent of the state of the *Digital Sync Format* and will remain as composite sync whether the **notCorHSync** and **notVSync** sync pins are being used to provide (in master mode) composite or separate sync, or (in slave mode) being driven with vertical or vertical and horizontal sync.

If two (or more) video sources are to be used in a video synchronous system, it may be necessary to only have one (master) device generate the synchronization pulses (whether composite or separate, and mixed with the DAC outputs or not). See the brief discussion in Section 10.6. The delay mentioned above from the master to the slave sync pulses must also be borne in mind when designing such a system, although this may be eliminated (or rather moved), depending upon the video timing specification, as discussed above.

10.5.2 SYNCHRONIZATION PULSE LEVELS

When synchronizing a CVC family device with another video source (whether as a master or a slave), the sync pulse levels may need to be checked. The CVCs drive their syncs at normal TTL levels (+5V, 0V), active low. Other video apparatus (such as video cameras and mixers) sometimes A.C. couple their sync signals, which can result in the signal still being driven active low, but actually varying about an arbitrary D.C. value, and may go negative with respect to the CVC power rails. Alternatively, some video systems drive sync signals from 0V to 1V or 2V (the sync pedestal level), in which case some pulse level shifting may be necessary, perhaps using comparators. In these situations, the sync signal levels may make obtaining video synchronization impossible unless the levels are compatible.

10.6 USE OF GENLOCK

Genlock derives horizontal and vertical sync pulses (and a pixel clock frequency) from a composite video source. Several devices are on the market for achieving this. A PLL is used to generate the pixel clock frequency from the horizontal sync pulses, using an appropriate clock multiplier or oscillator circuit. This allows one video source to be used as a master to several slave video sources, with guaranteed synchronization between the different video systems. Such schemes are commonly adopted in distributed studio graphics systems, where many different video sources may be used and mixed together. To use a CVC (as a slave) with a genlocked system, it is necessary to run in times one, external clock mode (not using the PLL), using the externally generated pixel clock, and synchronizing to the externally produced vertical (and horizontal if in interlaced mode) sync signals.

10.7 SUMMARY

10.7.1 INTERLACED AND NON-INTERLACED SLAVE MODE SYNCHRONIZATION

- Use of a common pixel clock (Genlock) eliminates the following two effects:
  - The need for synchronization to within one SClk period (4 not 1 pixel) using different pixel clock sources.
  - Beating of different pixel clock phases.
- The slave display will be delayed by *n* SClk periods with respect to the master as discussed in Section 10.4.5. Some or all of this delay can be moved from BackPorch to FrontPorch.
- Composite sync can be mixed onto the DAC outputs irrespective of the *Digital Sync Format*, whether separate or composite.
- CVC synchronization pulses are TTL levels.
- In master mode, VSync is always driven, regardless of the state of the *Digital Sync Format*. Note that the IMS G335 allows the sync outputs to be tristated.

10.7.2 NON-INTERLACED SLAVE MODE OPERATION

- An external VSync signal is required.
- *Digital Sync Format* may be separate or composite.
- *Analog Video Format* may be video only, or video and sync composite.

10.7.3 INTERLACED SLAVE MODE OPERATION

- External VSync and HSync signals are both required.
- No external HSync pulses should occur during frame flyback, except on the IMS G335 with the *Flyback Waveform* set to Hsync during frame flyback.
- *Digital Sync Format* must be separate.
- *Analog Video Format* may be video only, or video and sync composite.
- The position of an external HSync pulse with respect to an internally produced HSync pulse is important.

Table 10.1.

Slave Mode	Digital Sync Format	Analog Video Format	HSync Required	VSync Required
Non-Interlaced	Comp or Sep	As required	N	Y
Interlaced	Sep	As required	Y	Y

10.8 CONCLUSION

The CVC family of devices can be used in both master and slave modes. This allows multiple devices, or other video sources, to be used with video synchronization between them. Various different modes of operation, using interlacing and non-interlacing, using 1, 2, 4, 8, 15, 16 and 24 bits per pixel, and different resolutions can be used.

10.9 REFERENCE

1 *The Computer Graphics Databook*, 3rd edition, SGS-THOMSON, November 1992.

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

USING THE ON-CHIP SELF TEST CAPABILITIES OF CVCS

GRAPHICS APPLICATIONS GROUP, BRISTOL

11.1 INTRODUCTION

This Application Note applies to the following SGS-THOMSON Color Video Controllers (CVCs) and describes how features embodied in their design facilitate the testing of the devices and systems containing them:

- **IMS G332**
- **IMS G364**
- **IMS G335**

Throughout this document, information that is generic to all devices will be referred to by the collective name CVC. Where specific information pertains only to one or more particular devices, the actual name of the device(s) will be specified.

This document should be read in conjunction with the relevant datasheet for the particular CVC required.

11.2 IMS G3xx COLOR VIDEO CONTROLLERS

CVCs are dedicated support chips that provide all necessary functions for controlling the real time raster operation of raster scan video systems, using dual ported video DRAMs (VRAM). The facilities provided are designed to isolate the host processor from the constraints of the real time system without in any way interfering with ability of the processor to specify and manipulate screen data.

The devices consist of a programmable Video Timing Generator (VTG), a Framestore Manager with screen refresh and auto line increment capability, a triple 256 location 8-bit lookup table (LUT), a triple 8-bit video DAC and an on-chip phase-locked loop (PLL). CVCs also provide a hardware cursor and 16 and 24-bit color operation (G332 does not support 24-bit color).

11.3 THE NEED FOR TESTABILITY

One of the areas of most concern in the computer industry is the area of testability. The ability to accurately test machines accrues benefits to both the manufacturer and customer.

From the manufacturer's point of view, systems that have features embodied within them to facilitate testing, can usually be tested more quickly than those which do not. This reduces the manufacturing time with a resultant reduction in build cost and increase in manufacturing throughput.

Systems that feature built in self test features which are accessible by the system BIOS or device driver software can incorporate more thorough system diagnostics and power on self test procedures than would otherwise be possible. The ability to perform such diagnostic procedures elevates the customers perceptions about the overall quality of the machine.

It is however usually impractical to incorporate additional logic components on the printed circuit board specifically for test purposes, either because of real estate limitations, cost or the unavailability of test access points (if the required data path is not brought out from a device).

The video subsystem by virtue of its shear complexity and data rate imposes an enormous processing requirement upon the system test function. For VRAM based systems, the integrity of both the host CPU random access data path and the Serial Mode Access data path must be verified in order to ensure correct system operation.

11.4 TESTING MEMORIES

It is beyond the scope of this application note to thoroughly cover memory testing strategies, but we must at least cover some of the issues in order to maximize the utilization of the built in test facilities of the CVC.

To fully test a large memory array for all possible errors is deemed to be impossible as the number of permutations of address and data combinations is vast. However by taking into account a knowledge of the architecture of the memory system we can make sensible decisions as to which address and data combinations to test that may yield the highest confidence of satisfactory operation. Such features of the architecture to consider are:

- How many banks
- Interleaved or non interleaved
- Bank boundaries
- Word width
- RAS/CAS selection
- Volatility due to perhaps inadequate refresh rates

Thus in general the usual tests performed are:

- Walking 1s in the data 0x00000000, 0x00000001, 0x00000010, ... 0x10000000
- Check for stuck bits 0xAAAAAAAA, 0x55555555, 0xA5A5A5A5, 0x5A5A5A5A

- All set or reset 0xFFFFFFFF, 0x00000000
- Random access addressing tests to ensure that writing to a location in one bank does not affect other banks
- Writing random data to random locations
- Refresh tests

11.5 OVERVIEW OF TEST RELATED FEATURES

Where it is impractical for the host CPU to test features of the device because of real time video data rates, functionality has been added on-chip to permit the test of chip components that would otherwise be untestable.

The data path from the Serial Access Mode port (SAM) through to the Video DACs is not directly accessible from the host CPU. In any case the video data rate is such that it would be impossible for a host CPU to test the data path unaided.

To facilitate the testing of the video data path presented to the video DACs, three 24-bit checksum registers, one for each color channel, are provided as shown in Figure 11.1. They are implemented as Linear Feedback Shift Registers (LFSRs) using the Type II design which use internal Exclusive OR feedback taps (Figure 11.2). These accumulate over a frame period to produce, by the end of the frame, a checksum that is proportional to the current contents of the actively displayed region of the framestore, color palette, cursor attributes and the mask register.

In order to use these features, test software must first write suitable test patterns of the types previously discussed into the framestore. After writing each pattern, the test software should compute the expected checksum that should be generated by the checksum registers at the end of the frame. In frame flyback time, the test software should then read the contents of the three flyback registers. If the frame flyback detection is connected to the interrupt line of the processor, an interrupt service routine should perform the reads of the checksum registers. For systems that do not use the interrupt line, the processor should poll **FrameInactive**, the frame flyback signal from the CVC. To ensure that the sampling of the checksum registers is carried out entirely in the frame flyback period, the test software should perform the algorithm shown below.

The reading of the registers must take place before the falling edge of **FrameInactive**. It is permissible to read each checksum register in separate frame flyback periods, providing the content of the frame store does not change.

```
WHILE NOT Frameinactive
  ... Do nothing, since the frame is currently being output to the DACs
/* We are now at the start of Frameinactive */
read_checksum_register(red);
read_checksum_register(green);
read_checksum_register(blue);
```

11.5.1 CHECKSUM TEST

It is also possible to test the functionality of the checksum registers themselves independently of the pixel data. This is achieved by using the checksum test facility which implements a checksum test by forcing zero to be applied to the inputs of each checksum register. The effect of this is to test the checksum registers themselves prior to further system diagnostics. In order to invoke the checksum test option, bit [8] of Control Register B should be set to 1.

Figure 11.1. IMS G335 Block Diagram showing checksum registers

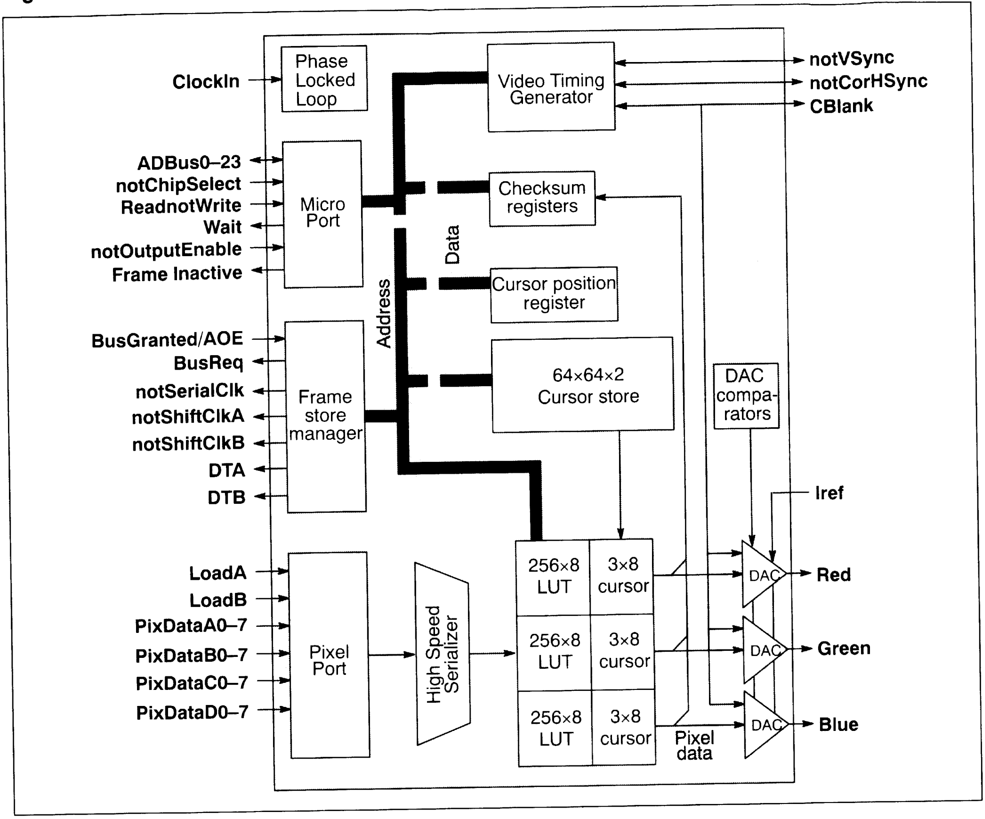
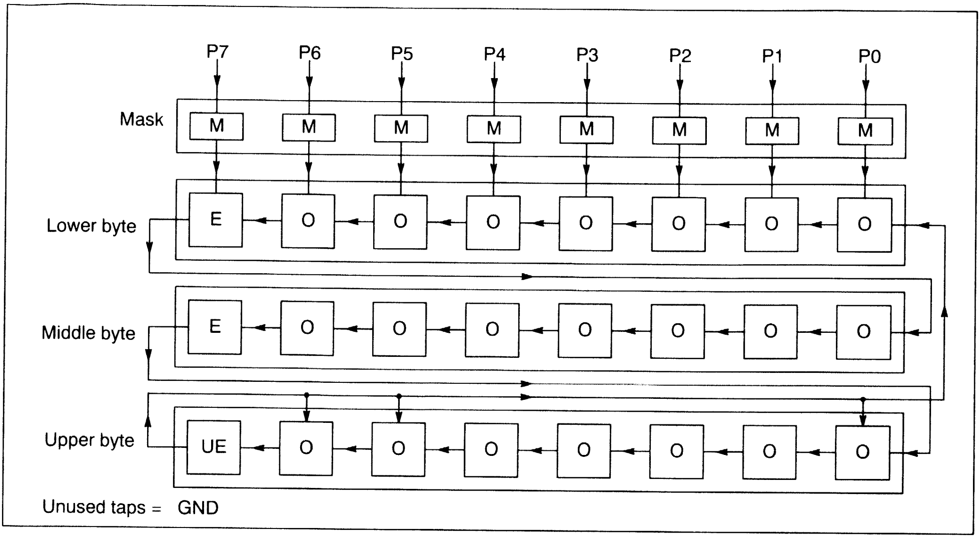


Figure 11.2. IMS G335 checksum register



11.6 PROGRAMMING CONSIDERATIONS

The test software to invoke this test functionality should implement the following functions (assuming 8-bit frame stores):

```
disable hardware cursor;
for each test pattern (0x00, 0x55, 0xAA, 0x5A, 0xA5, 0xFF)
  {
    — first test random port
    write test pattern to every location in the frame store;
    read back each location in the frame store and compare with test pattern;
    if random access port ok
      {
        calculate the check sums using the contents of the frame store;
        wait for frame flyback;
        read the checksum registers;
        compare the checksum calculated with the contents read from the
           checksum registers;
      }
  }
```

In line with the recommendations made in the section on testing memories, it is also suggested that random data be written to the framestore to test for errors in bank switching on the serial access ports.

A framestore which has every location the same will still generate a valid checksum even when incorrect banks are being transferred to the CVC serial port.

11.7 CHECKSUM CALCULATION

The following algorithm should be used to calculate the expected contents of the checksum registers for the defined frame store. Note this algorithm assumes that the cursor is disabled, the mask register is set to **0xFFFFFF** and that interlaced mode is not being used (this however does not preclude the developer from modifying the algorithm to take these features into account).

```
void calculate_chksums (long int results [3],
                        int screen_width, int screen_height,
                        char palette [256][3])
{
  int color, x, y;
  for (color = 0; color < 3; color++) /* reset checksums */
  {
    results [color] = 0x0FFFFFFL;
  }
  for (y = 0; y < screen_height; y++)
  {
    for (x = 0; x < screen_width; x++)
    {
      char pixel;
      if (on_screen)      /* valid pixel */
        pixel = read_pixel (x, y);
      else                /* else border pixel */
        pixel = 0;
      /* We must calculate check sums for red, green and blue */
      for (color = 0; color < 3; color++)
      {
        results [color] ^= (long) palette [(int) pixel] [color];
        results [color] <<= 1;
      /* Check if MSB is zero */
        if ((results [color] & 0x01000000L) == 0)
        {
      /* Yes! therfore set LSB to 1 and feedback to taps */
          results [color] |= 1;
          results [color] ^= (long) 0x0c20000L;
        }
      /* Truncate checksum to 24 bits */
        results [color] &= (long) 0x0FFFFFFL;
      }
    }
  }
}
```

11.8 CONCLUSION

This Application Note has described how the provision of built in self test hardware within CVCs facilitates the testing of the serial access data path from the VRAMs to the output of the color palette. This enables the system designer to invoke these capabilities within the overall system test strategy.

11.9 REFERENCES

1 *The Graphics Databook*, 2nd edition, INMOS Ltd, 1990

2 *The Computer Graphics Databook*, 3rd edition, SGS-THOMSON, 1992

3 *Digital Systems Testing and Testable Design*, Abramovici, Breuer and Friedman

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

SOME COMMON PROBLEMS DRIVING MONITORS WITH CVCS

GRAPHICS APPLICATIONS GROUP, BRISTOL

12.1 SOME COMMON PROBLEMS DRIVING MONITORS WITH CVCS

This note details some common problems and possible solutions when driving monitors with Color Video Controllers (CVCs). The effect of all the problems described is to cause the monitor screen to blank.

12.1.1 VIDEO SYNC TIMING OUT OF SPECIFICATION

Some monitors blank if the video timing being used to drive them is out of their acceptable tolerance. This is not always obvious, as there will clearly be nothing displayed on the screen. The solution is to recheck the video timings and video frequency being used.

12.1.2 MULTIPLE SYNC SIGNALS

It is sometimes possible to inadvertently connect both the DAC outputs and the sync outputs to a monitor. The operation of the majority of monitors will not normally be affected. With some monitors, however, this can cause problems because the CVC could be driving composite sync on video, composite sync on **notCorHSync**, as well as vertical sync on **notVSync**. The monitor is thus receiving three forms of vertical sync signal and two forms of horizontal sync signal. If this causes a problem, one solution is to select video only on the DACs and separate digital sync on the sync outputs. Alternatively (on the IMS G335), the sync outputs can be tristated (Control Register B, bit 9).

12.1.3 HSYNC DURING FRAME FLYBACK

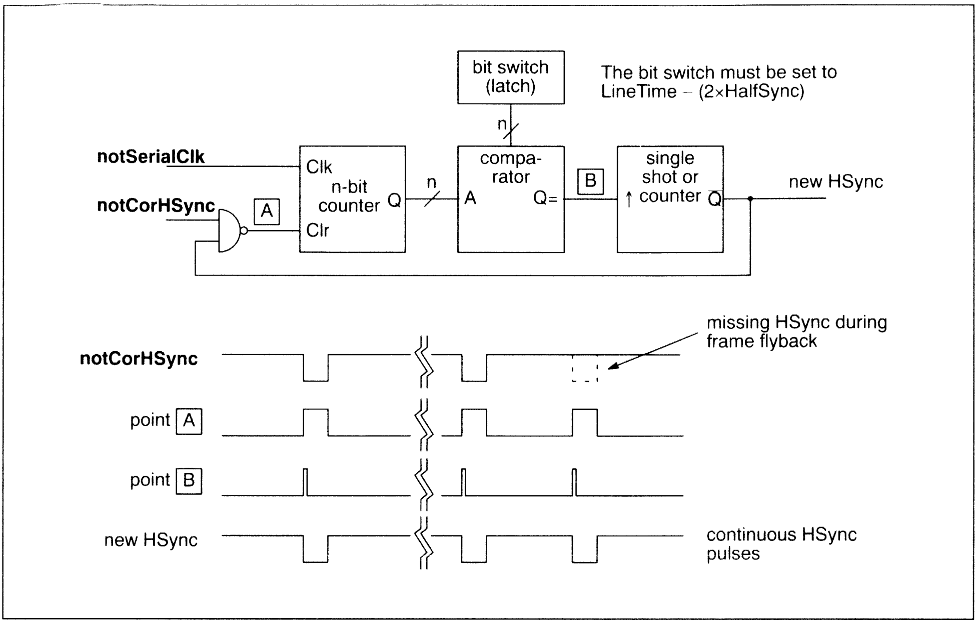
The IMS G335 can be programmed to produce HSync during frame flyback. This option is not available on the IMS G332 and G364. Where HSync is not present during frame flyback, problems can occur with certain monitors and scan rate converters that rely on continuous HSync to keep a phase locked loop in synchronization. The obvious solution for new designs is to use the IMS G335. For current designs, an external circuit is required.

One way of generating the missing HSync pulses during frame flyback is to use a counter and a single shot (or a second counter) as shown in Figure 12.1.

The counter counts notSerialClock pulses and is reset by HSync or the comparator output going high. The output of this counter is compared against a number n (conveniently represented in Figure 12.1 by a bit switch but could be programmed into a latch). The value n is Linetime – (2× HalfSync), i.e. n represents the HSync inactive (high) time. When the counter output equals n, the comparator output is used to trigger a single shot monostable device which resets the counter. The period of the monostable is adjusted to be equal to the required HSync pulse width (twice HalfSync). Alternatively, the comparator can start a second counter which resets itself when it has counted twice HalfSync.

The output of the single shot is thus a continuous HSync pulse stream. Resetting the counter on both HSyncs ensures that the output HSync pulse stream is kept in correct synchronization and phase with the original HSync pulses.

Figure 12.1. Circuit example for generating HSync pulses during frame flyback



GRAPHICS GLOSSARY

GRAPHICS GLOSSARY

The following is a glossary of some frequently used terms in this book, together with some other more general terms relevant to graphics systems. Descriptions are given relating specifically to graphics systems rather than their more general use. We hope this will be useful for you.

Analog A signal whose value can vary continuously over its known range, i.e. it is not limited to predetermined levels (compare to digital signal). An analog signal is used to drive each color gun on an RGB monitor.

Anti-sparkle A feature of Palette-DACs which allows access to the palette contents during active video without disruption to screen contents. It works by replicating the last pixel displayed during the palette access. Devices without this feature will 'lose' a pixel during such accesses, and repeated accesses during active video will cause an unwanted 'snowing' or 'sparkling' effect on the screen.

Aspect ratio The ratio of width to height, usually of a display screen. An emerging workstation standard is 1280×1024 pixels (i.e. 5:4 aspect ratio). The aspect ratio is 4:3 for broadcast television pictures.

Back porch The part of the horizontal blanking pulse that comes between the trailing edge of the horizontal synchronising pulse and the start of the displayed line.

Bandwidth The maximum rate of data transfer between two points in a system. Usually measured in Mbytes/second.

BIOS Basic Input Output Sytem. An element of the DOS operating system on PC computers, the BIOS is responsible for handling the details of input/output operations. At its lowest level, the BIOS contains routines called device drivers tailored to the specific requirements of each peripheral device.

Bit map The digital representation of an image as rows and columns of pixels, usually stored in the video memory (RAM). Each pixel can be represented by one bit (monochrome) or up to 24 bits (photorealistic color) of data. The pixels in the bit map correspond directly to the appropriate on-screen location.

Black level The amplitude of the analog video signal which defines the level below which the electron beam representing that color in the monitor is not visible.

Blanking A composite video signal comprises a range of values used for picture representation and a range used for synchronizing information. Blanking is a method where a certain level is defined (Blanking level), below which the signal is interpreted as synchronizing information, and above is picture information.

Blanking period The time during which the composite video signal is reduced in amplitude to the blanking level, i.e. below which the display electron beam is cut off. This allows the beam to go from the right side of the display to the left at the end of each linescan (horizontal blanking) and from the bottom right of the screen to the top left of the screen (vertical blanking) without being visible.

CLUT Color Look-Up Table. This is the high-speed RAM on many graphics devices which is used to store color values for display. In pseudo color modes the pixel data is used as an address reference to this RAM. Also known as the Color Palette or LUT only.

Color value A word stored in the Look-Up Table defining the color of a pixel in terms of its red, green and blue intensities.

Color expansion A method of color display where an address (up to 8 bits per pixel) is stored in memory, and used as a reference to a wider color word stored in a look-up table. This data 'expansion' of color representation saves significantly on memory requirement.

Composite sync All monitors require two types of synchronization pulses to provide a timing reference indicating the point where a new frame or a new line starts. These are called the frame sync and line sync respectively. These timing signals are distinguished by their duration (frame sync pulses are longer than line sync pulses) and can therefore be mixed as a single signal, known as composite sync. (They can also be mixed with the picture information, see Composite Video.) For a typical high resolution monitor, a frame sync pulse might be 50 to 100ms wide, whereas line sync pulses would be in the region of 2 to 6μs.

Composite video There are two different approaches to providing sync pulses. Composite video superimposes these timing pulses onto the analog video signal containing the picture information (approximately a 0.3V sync pedestal on a 0.7V video signal if using the RS170 standard). The monitor is then able to strip the sync signals off for timing control. A separate sync video system, as its name suggests, supplies the sync pulses on an extra signal line to the monitor (having the advantage of not requiring a circuit to strip the sync timing from the video inputs inside the monitor). A composite video signal is therefore the complete video signal consisting of the picture information, horizontal and vertical blanking, and frame and line synchronization.

CRT Cathode Ray Tube. The evacuated glass display device forming the display of most TVs and computer monitors.

CVC Color Video Controller. A family of high performance devices from SGS-THOMSON which integrate all the control functions required to produce high resolution color graphics displays. A CVC is a complete graphics subsystem on a chip.

DAC Digital to Analog Converter. Three DACs are found on all SGS-THOMSON graphics devices, one for each color channel Red, Green and Blue.

Direct Color An alternative description of True Color. Pixel data reaches the graphics device and is applied 'direct' to the DACs, i.e. bypassing the look-up table. This increases the number of simultaneous colors, but also increases the amount of memory required to store the display data.

DRAM refresh The operation of maintaining data stored in dynamic RAMs. The data stored in a DRAM is represented by charges across a grid of capacitive cells. The charges leak over time so it must be refreshed periodically. This is different from screen refresh, which occurs to maintain a continuous display on the screen.

EGA Enhanced Graphics Adaptor. Early PC graphics standard defined by IBM which offered up to 640 x 350 display resolution with up to 16 colors. The PC graphics standard has since been replaced by VGA and more lately accelerated AVGA solutions.

Flicker The perceived fluctuations in display intensity of a screen, which can cause eye-strain if viewed for long periods. Can be reduced by increasing the persistence of the phosphors in the monitor or by increasing the frame refresh rate.

Fly-back The period when the blanked electron beam returns from the right side of the screen to the left, or from the bottom of the screen to the top. This period is often used to update the palette or toggle the screen etc.

Frame grabber A device used to take an analog TV picture and digitize it into discrete pixels for transfer to and possible storage in computer memory.

Frame refresh rate The number of times per second that the frame contents are redrawn on the screen. Usually between 50 and 80Hz. Too low a frame rate can lead to a perception of the image flickering on the screen. (See 'Flicker'). ISO standards are pushing frame rates higher (currently 72Hz) to meet ergonomic requirements of users. A higher frame rate requires a corresponding higher pixel rate.

Framestore (or Frame buffer) A portion of system memory containing data which represents the displayable pixels. The frame buffer may contain more pixels than can be displayed simultaneously on the screen, allowing the display 'window' to pan and scroll around the total frame buffer. A system may use multiple frame buffers and switch between them for different applications, or to achieve animation effects. Also referred to as 'Bit Map'.

Framestore Manager High resolution displays now require Video RAM to achieve the high pixel rates required. The Framestore Manager is a feature on the IMS G3xx family of Color Video Controllers which provides VRAM serial port clocking, SAM refresh addresses and SAM transfer signals to maintain a constant supply of data from VRAM to the pixel ports. This allows a packed pixel framestore of any pixel depth to be supported, regardless of screen format required.

Front Porch The portion of a horizontal blanking pulse that comes between the end of the displayed line and the leading edge of the horizontal sync pulse.

Full Color As True Color, but usually referring specifically to 24 bits per pixel. In a full color system color expansion is NOT used, and display is from a choice of over 16 million colors. It is used to create photorealistic images, but is expensive in memory requirement.

Gamma correction Due to the nature of the phosphors used on color monitor screens, color intensity displayed is not a linear function of the applied electron beam energy. Gamma correction is a method of eliminating these non-linearities by passing the pixels through a look-up table and applying a precalculated correction factor before applying to screen.

Grey scale A series of light intensities incrementing from zero to full scale where RGB components are always equal. The larger the number of discrete steps in this series, the higher the grey scale value.

GUI Graphical User Interface. A computer control system that allows the user to enter commands by 'pointing and clicking' on a graphical background, rather than typing in ASCII character commands. Although GUIs can provide improved productivity and ease-of-use, they require considerably higher performance from the graphics subsystem to operate effectively.

HDTV High Definition TeleVision. A proposed standard for improved quality broadcast TV. Competing proposals from different regions have resulted in no agreed standard as yet. Most proposals go for 16:9 aspect ratio and doubling the number of lines.

High Color Term used to describe EITHER (as implemented by IMS G173/G174) a method for building larger pixel words from consecutive bytes applied to an 8-bit VGA compatible pixel port, OR the same as true color, but usually restricted to 15-23 bits-per-pixel.

Horizontal sync The synchronizing signal that indicates the end of a display line on a CRT and enables the retrace of the electron beam back across to the beginning of the next line.

Interlaced scanning A system of picture scanning where odd numbered lines, which make up an odd field, are interlaced with the even numbered lines of an even field. These two fields are refreshed alternatively. The two interlaced fields constitute one frame. The advantage of this system is that it requires a much lower pixel rate to achieve the same resolution as a corresponding non-interlaced picture. The disadvantage is that an interlaced display tends to flicker more.

Interleaving In high speed graphics systems, it is not always possible to supply data fast enough from a single bank of VRAMs. Interleaving is a method which allows two separate banks of VRAM to be used, each running at half the frequency required when using a single bank. Data is then loaded alternately from one bank then the other. Two sets of clocking control signals are required (as provided by the CVC family of products).

Jitter Short-term variations in the amplitude, phase or frequency of a signal. The pixel rate clock signal is most critical in graphics systems, since any jitter on this will cause momentary displacements on the screen, giving a shaky or 'jittery' appearance. The high speed clock signal used by CVC products is generated by an on-chip PLL, which has exceptionally high stability and is virtually jitter-free.

Line rate The total number of picture lines per second. Can be calculated from (Lines per frame × frame refresh rate). Usually in the order of 20-100kHz.

Look-Up Table (LUT) See CLUT.

Multiplexing This is a method applied by a CVC to the pixel data stream to achieve the high data rates required. A wide pixel word (32 or 64 bits) is input from the VRAM framestore, clocked in at a low speed which can be driven comfortably at TTL rates. The word is then split up into individual pixels and serialized internally by the CVC to the full video rate. Pixel multiplexing keeps all external signal rates low, so easing system design and greatly assisting emissions compliance.

Multisync monitor A monitor that is able to detect and adjust to the frequency of video signal that is being applied to it. Compare to a fixed frequency monitor which is only able to operate at one or a number of specified frequencies.

Non-interlaced scanning A picture scanning method where every line of the display is refreshed on every frame update. Compare to interlaced scanning. Non-interlaced scanning requires higher pixel rates, but produces a more stable display.

NTSC National Television Standards Committee. A group instrumental in setting standards in North America and Japan, e.g. RS-343A, EIA-343A and RS-170. The NTSC signal format used for TV broadcast and display has 525 lines at 60Hz. CVCs are able to drive video signals to conform with the NTSC timing standard.

PAL Phase Alternation Line. A signal format standard used for TV broadcast and display in Europe. It operates at 625 lines at 50Hz field rate. CVCs are able to drive video signals to conform with the PAL timing standard.

Palette See CLUT.

Panning The ability to move part (or whole) of the display area with respect to the left or right edge of the screen to view a part of the total image which is off-screen.

Pixel Picture Element. The smallest unit of display memory which can be individually controlled, therefore corresponding to the area of smallest detail that can be resolved on the screen. Is simple terms, if a picture is made up of a series of lines, and each line is made up of a series of dots; a pixel is a dot.

Pixel depth The number of bits used to describe each pixel. This determines the number of different colors that may be displayed simultaneously.

Pixel rate The rate at which pixels are drawn on the screen during active video periods. In low resolution systems this is roughly equal to (horizontal resolution × vertical resolution × frame rate). However, as resolution increases, proportionally more time is required for line and frame flyback, thus there is less time for drawing. This demands a higher pixel rate during actual drawing periods. Pixel rates range from 27MHz for 640 × 480 at 60Hz, through 66MHz for 800 × 600 at 70Hz, to 135MHz for 1280 × 1024 at 75Hz.

Presentation Manager The windowing GUI from IBM, running under OS/2.

Pseudo color A scheme where the pixels experience color expansion when passed through a look-up table prior to being sent to the video DACs. The pixels values in the framestore are used as an address (typically 1 to 8 bits) which points to a location in the look-up table holding a color value described by a much larger number of bits (typically 18 or 24 bits). This color value is then applied to the DACs. This greatly reduces the amount of frame store required to hold an image while still retaining the total color resolution.

RAMDAC A name used to describe a family of devices with look-up table RAM and triple DACs. Also known as CLUTs, color palettes, graphics DACs, or Palette-DACs.

Raster scan The grid pattern traced by the electron beam on the face of the CRT in a TV or similar raster-scan display device which illuminates the phosphor surface and creates a displayed image.

Refresh rate See Frame Refresh Rate.

Resolution The screen resolution refers to the number of individual pixels making up the screen, and is made up of horizontal resolution (pixels/line) and vertical resolution (lines/screen.) The color resolution refers to the total possible number of colors which are available simultaneously, and is determined by the pixel depth. The DAC resolution refers to the number n, where 2^n is the number of discrete analog levels which the DAC is capable of converting.

RGB signal The analog signals required to drive the Red, Green and Blue color guns of an analog monitor.

Retrace The line traced by the scanning beam of a picture tube as it travels from the end of one horizontal line to the beginning of the next (horizontal retrace) or from the end of one frame to the beginning of the next (vertical retrace.)

SAM Serial Access Memory. The serial shift register component of a video RAM device. During a VRAM transfer cycle a newly addressed row of data is transferred from the RAM part of the device to update the SAM. This data is then streamed synchronously to the monitor display via a Palette-DAC or CVC (see Video RAM).

Screen format A term used to describe the screen contents in terms of its horizontal and vertical resolution and bits-per-pixel.

Scrolling The ability to move part (or whole) of the display area with respect to the top or bottom edge of the screen to view a part of the total image which is off-screen. This can be achieved very easily with the CVC family of devices simply by changing the TopOfScreen register value in the VTG.

Split SAM Split SAM video RAM devices have their serial shift register split into two, one half being updated while the other is shifting out. The use of split SAM devices makes VRAM update timings much less critical and simplifies system design.

True Color A scheme where the pixel data bypasses the look-up table (except for gamma correction) and is sent straight to the DACs without color expansion. This method allows access to a much larger range of colors simultaneously, necessary for smooth shading and photorealistic images, but also requires significantly more memory than pseudo color.

Vertical sync The synchronizing signal that indicates the end of a frame of display on a CRT and enables vertical retrace of the electron beam back up to the top left-hand corner of the screen.

VGA Video Graphics Array. The graphics standard introduced by IBM in their PS/2 range of PCs, and by far the dominant PC graphics standard shipping today. Standard VGA offers up to 640 x 480 resolution, and up to 256 colors. The basic standard has been improved or enhanced many times by 3rd parties, with SuperVGA offering up to 1024 x 768 with up to 65,000 colors. IBM use the IMS G171 Palette-DAC throughout the PS/2 range to achieve the VGA standard, and the IMS G176 is the de facto industry standard Palette-DAC for all VGA systems.

Video memory See framestore.

Video RAM or Video DRAM Video Random Access Memory. A dual-ported memory device designed specifically for computer graphics applications. One port is a standard random access port which can be updated asynchronously by the processor. The other port consists of a long serial shift register which allows data to be streamed synchronously to the monitor display via a Palette-DAC or CVC. Synchronous register reloads allow a constant datastream to be maintained.

VTG Video Timing Generator. The logic which provides the synchronized horizontal and vertical sync timing signals to control a monitor. In the CVC family, the VTG is provided on-chip, and is a highly programmable state machine, capable of producing the appropriate signals to drive almost any monitor as well as TV standard signals for NTSC and PAL.

Windows (TM) The windowing GUI from Microsoft running under DOS.

ORDERING INFORMATION

ORDERING INFORMATION

PALETTE-DACS

These devices are used at the output stage to drive an analog monitor for VGA, SuperVGA, 8514/A and High Color graphics display systems. They comprise triple Digital to Analog Converters, high speed Look-Up Table RAM and microprocessor interface.

Description	Max. clock rate (MHz)	Package	Part Number
6-bit High Color Palette-DAC with PixMix	80	44-pin PLCC	IMS G173J–80Z
8-bit True Color Palette-DAC with PixMix	85	44-pin PLCC	IMS G174J–85Z
	100	"	IMS G174J–10Z
6-bit VGA Palette-DAC with anti-sparkle	66	28-pin DIP	IMS G176P–66SF
	80	"	IMS G176P–80SF
	66	32-pin PLCC	IMS G176J–66SF
	80	"	IMS G176J–80SF
	66	44-pin PLCC	IMS G176J–66ZF
	80	"	IMS G176J–80ZF
6-bit Low Power VGA Palette-DAC	50	32-pin PLCC	IMS G177J–50S
8-bit True Color Palette-DAC with 16-bit pixel port	135	44-pin PLCC	STG 1700J–13Z

COLOR VIDEO CONTROLLERS

These devices are highly integrated general purpose graphics chips. They integrate a Palette-DAC with all the control functions required to transfer data from video memory to the display monitor. They are a complete high performance graphics subsystem on a chip.

Description	Max. clock rate (MHz)	Package	Part Number
Color Video Controller with programmable VTG, PLL, VRAM framestore manager, 32-bit MUXed pixel inputs, hardware cursor, 1, 2, 4, 8, 15, 16 bits per pixel. *Not recommended for new designs*	85	100-pin CQFP	IMS G332F–85S
	100	"	IMS G332F–10S
	110	"	IMS G332F–11S
Color Video Controller with programmable VTG, PLL. VRAM framestore manager, 64-bit MUXed pixel inputs, hardware cursor, 1,2,4,8, 15, 16, 24 bits per pixel. *Not recommended for new designs*	85	132-pin PGA	IMS G364G–85S
	100	"	IMS G364G–10S
	110	"	IMS G364G–11S
NEW Color Video Controller, programmable VTG, PLL, VRAM framestore manager, split SAM support, 32-bit MUXed inputs, hardware cursor, 1,2,4,8,15,16,24 bits per pixel.	110	100-pin CQFP	IMS G335F–11S
	135	"	IMS G335F–13S

Graphics FAX

Thank you for your interest in SGS-THOMSON graphics products. If you wish to receive further information or technical assistance on any of our products, please mail or fax this page to your local SGS-THOMSON sales office.

Name: ______________________

Job title: ______________________

Company: ______________________

Company business: ______________________

Address: ______________________

____

____

Phone: ______________________

Are you currently using our products?

□ Yes □ No

What graphics applications are you involved with?

□ VGA graphics systems

□ High performance PC boards

□ MAC add-in boards

□ Workstations

□ X-terminals

□ Multimedia

□ Imaging systems

□ Video products

□ Other: ______________________

What graphics performance do you require?

Resolution__________ × __________

Frame rate__________ Hz

Color resolution:–

□ 8-bit □ 15/16-bit □ 24-bit □ 32-bit

What graphics controller/coprocessor do you currently use? ______________

□ I would like a salesman to call me

□ I would like an application engineer to call me

□ Please send me datasheets on:

□ Palette-DACs

□ Color Video Controllers

□ Please put me on your mailing list for product updates

Thank-you. All information is treated in confidence

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

8011 GRASBRUNN
Bretonischer Ring 4
Postfach 1122
Tel.: (49-89) 460060
Telefax: (49-89) 4605454
Teletex: 897107=STDISTR

6000 FRANKFURT
Gutleutstrasse 322
Tel. (49-69) 237492-3
Telefax: (49-69) 231957
Teletex: 6997689=STVBF

3000 HANNOVER 51
Rotenburger Strasse 28A
Tel. (49-511) 615960-3
Teletex: 5118418 CSFBEH
Telefax: (49-511) 6151243

8500 NÜRNBERG 20
Erlenstegenstrasse, 72
Tel.: (49-911) 59893-0
Telefax: (49-911) 5980701

7000 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

08021 BARCELONA
Calle Platon, 6 4th Floor, 5th Door
Tel. (34-3) 4143300-4143361
Telefax: (34-3) 2021461

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

U.S.A.

NORTH & SOUTH AMERICAN
MARKETING HEADQUARTERS
1000 East Bell Road
Phoenix, AZ 85022
(1-602) 867-6100

SALES COVERAGE BY STATE

ALABAMA
Huntsville - (205) 533-5995

ARIZONA
Phoenix - (602) 867-6217

CALIFORNIA
Santa Ana - (714) 957-6018
San Jose - (408) 452-8585

COLORADO
Boulder (303) 449-9000

ILLINOIS
Schaumburg - (708) 517-1890

INDIANA
Kokomo - (317) 455-3500

MASSACHUSETTS
Lincoln - (617) 259-0300

MICHIGAN
Livonia - (313) 953-1700

NEW JERSEY
Voorhees - (609) 772-6222

NEW YORK
Poughkeepsie - (914) 454-8813

NORTH CAROLINA
Raleigh - (919) 787-6555

TEXAS
Carrollton - (214) 466-8844

FOR RF AND MICROWAVE POWER TRANSISTORS CONTACT
THE FOLLOWING REGIONAL OFFICE IN THE U.S.A.

PENNSYLVANIA
Montgomeryville - (215) 361-6400

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 110001
LiasonOffice
62, Upper Ground Floor
World Trade Centre
Barakhamba Lane
Tel. (91-11) 6424603
Telex: 031-66816 STMI IN
Telefax: (91-11) 6424615

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
12th Floor
325, Section 1 Tun Hua South Road
Tel. (886-2) 755-4111
Telex: 10310 ESGIE TW
Telefax: (886-2) 755-4008

JAPAN

TOKYO 108
Nisseki - Takanawa Bld. 4F
2-18-10 Takanawa
Minato-Ku
Tel. (81-3) 3280-4121
Telefax: (81-3) 3280-4131

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of SGS-THOMSON Microelectronics.

SGS-THOMSON Microelectronics GROUP OF COMPANIES
Australia - Brazil - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - United Kingdom - U.S.A.

Recycled and chlorine free paper