Z80 CPU
2nd EDITION
ISSUED MAY 1982

CONTENTS

Execution Times

The execution time (E.T.) for each instruction is given in microseconds for an assumed 4 MHz clock. Total machine cycles (M) are indicated with total clock periods (T States). Also indicated are the number of T States for each M cycle. For example:

M CYCLES: 2 T STATES: 7 (4,3) 4 MHz E.T.: 1.75

indicates that the instruction consists of 2 machine cycles. The first cycle contains 4 clock periods (T States). The second cycle contains 3 clock periods for a total of 7 clock periods or T States. The instruction will execute in 1.75 microseconds.
Register format is shown for each instruction with the most significant bit to the left and the least significant bit to the right.

Z80® PROGRAMMING MANUAL

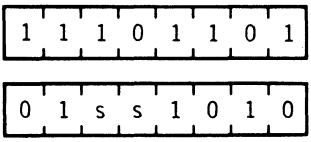
This manual contains a complete statement of the Z80 CPU instruction set. Each instruction is described by the operation, assembler mnemonic format op code. A description of the instruction operation is followed by the execution time with M cycles and T states, the flag status and, finally, an example of the instruction operation.
SGS will be grateful to receive from users information about any errors for points that are not clear in this manual which can be corrected in future editions.

ADC HL,ss

Operation: HL←HL + ss + CY

Format:

Opcode	Operands
ADC	HL,ss



Description:

The contents of register pair ss (any of register pairs BC, DE, HL or SP) are added with the Carry Flag (C flag in the F register) to the contents of register pair HL, and the result is stored in HL. Operand ss is specified as follows in the assembled object code.

Register Pair	ss
BC	00
DE	01
HL	10
SP	11

M CYCLES: 4 T STATES: 15 (4,4,4,3) 4 MHz E.T.: 3.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if carry out of Bit 11; reset otherwise
P/V:	Set if overflow; reset otherwise
N:	Reset
C:	Set if carry from Bit 15; reset otherwise

Example:

If the register pair BC contains 2222H, register pair HL contains 5437H and the Carry Flag is set, after the execution of

```
ADC HL,BC
```

the contents of HL will be 765AH.

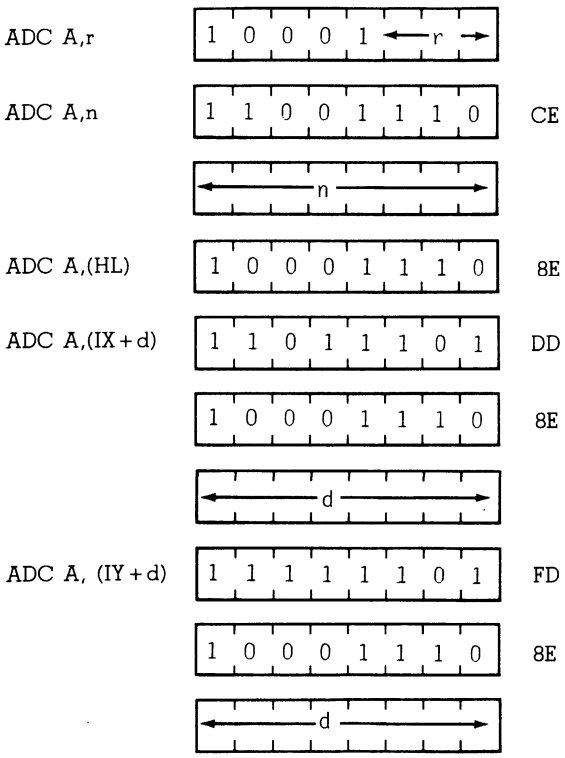
ADC A,s

Operation: A←A+s+CY

Format:

Opcode	Operands
ADC	A,s

The s operand is any of r,n,(HL), (IX+d) or (IY+d) as defined for the analogous ADD instruction. These various possible opcode-operand combinations are assembled as follows in the object code:



r identifies registers B,C,D,E,H,L or A assembled as follows in the object code field above:

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

The s operand, along with the Carry Flag (C in the F register) is added to the contents of the Accumulator, and the result is stored in the Accumulator.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
ADC A,r	1	4	1.00
ADC A,n	2	7 (4,3)	1.75
ADC A, (HL)	2	7 (4,3)	1.75
ADC A, (IX+d)	5	19 (4,4,3,5,3)	4.75
ADC A, (IY+d)	5	19 (4,4,3,5,3)	4.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if carry from Bit 3; reset otherwise
P/V:	Set if overflow; reset otherwise
N:	Reset
C:	Set if carry from Bit 7; reset otherwise

Example:

If the Accumulator contains 16H, the Carry Flag is set, the HL register pair contains 6666H, and address 6666H contains 10H, after the execution of

```
ADC A, (HL)
```

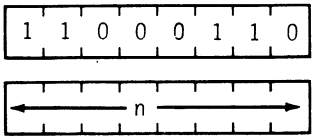
the Accumulator will contain 27H.

ADD A,n

Operation: A←A+n

Format:

Opcode	Operands
ADD	A,n



Description:

The integer n is added to the contents of the Accumulator and the results are stored in the Accumulator.

M CYCLES: 2 T STATES: 6 (4,3) 4 MHz E.T.: 1.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if carry from Bit 3; reset otherwise
P/V:	Set if overflow; reset otherwise
N:	Reset
C:	Set if carry from Bit 7; reset otherwise

Example:

If the contents of the Accumulator are 23H, after the execution of

```
ADD A,33H
```

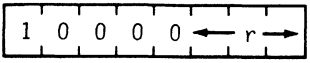
the contents of the Accumulator will be 56H.

ADD A,r

Operation: A←A+r

Format:

Opcode	Operands
ADD	A,r



Description:

The contents of register r are added to the contents of the Accumulator, and the result is stored in the Accumulator. The symbol r identifies the registers A,B,C,D,E,H or L assembled as follows in the object code:

Register	r
A	111
B	000
C	001
D	010
E	011
H	100
L	101

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Conditions Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if carry from Bit 3; reset otherwise
P/V:	Set if overflow; reset otherwise
N:	Reset
C:	Set if carry from Bit 7; reset otherwise

Example:

If the contents of the Accumulator are 44H, and the contents of register C are 11H, after the execution of

```
ADD A,C
```

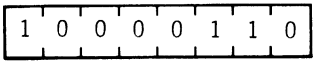
the contents of the Accumulator will be 55H.

ADD A,(HL)

Operation: A←A + (HL)

Format:

Opcode	Operands
ADD	A,(HL)



86

Description:

The byte at the memory address specified by the contents of the HL register pair is added to the contents of the Accumulator and the result is stored in the Accumulator.

M CYCLES: 2 T STATES: 7 (4,3) 4 MHz E.T.: 1.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if carry from Bit 3; reset otherwise
P/V:	Set if overflow; reset otherwise
N:	Reset
C:	Set if carry from Bit 7; reset otherwise

Example:

If the contents of the Accumulator are A0H, and the content of the register pair HL is 2323H, and memory location 2323H contains byte 08H, after the execution of

ADD A,(HL)

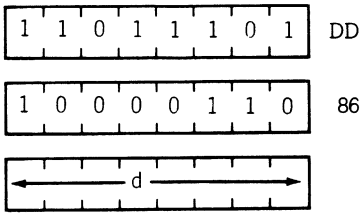
the Accumulator will contain A8H.

ADD A,(IX+d)

Operation: A←A + (IX + d)

Format:

Opcode	Operands
ADD	A,(IX + d)



Description:

The contents of the Index Register (register pair IX) is added to a displacement d to point to an address in memory. The contents of this address is then added to the contents of the Accumulator and the result is stored in the Accumulator.

M CYCLES: 5 STATES: 19(4,4,3,5,3) 4 MHz E.T.: 4.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if carry from Bit 3; reset otherwise
P/V:	Set if overflow; reset otherwise
N:	Reset
C:	Set if carry from Bit 7; reset otherwise

Example:

If the Accumulator contents are 11H, the Index Register IX contains 1000H, and if the content of memory location 1005H is 22H, after the execution of

ADD A,(IX + 5H)

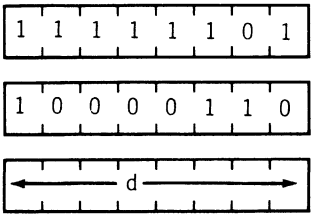
the contents of the Accumulator will be 33H.

ADD A,(IY+d)

Operation: A← A + (IY + d)

Format:

Opcode	Operands
ADD	A,(IY + d)



Description:

The contents of the Index Register (register pair IY) is added to a displacement d to point to an address in memory. The contents of this address is then added to the contents of the Accumulator and the result is stored in the Accumulator.

M CYCLES: 5 T STATES: 19(4,4,3,5,3) 4 MHz E.T.: 4.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if carry from Bit 3; reset otherwise
P/V:	Set if overflow; reset otherwise
N:	Reset
C:	Set if carry from bit 7; reset otherwise

Example:

If the Accumulator contents are 11H, the Index Register pair IY contains 1000H, and if the content of memory location 1005H is 22H, after the execution of

ADD A,(IY + 5H)

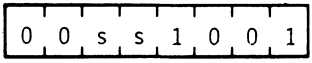
the contents of the Accumulator will be 33H.

ADD HL,ss

Operation: HL←HL + ss

Format:

Opcode	Operands
ADD	HL,ss



Description:

The contents of register pair ss (any of register BC,DE,HL or SP) are added to the contents of register pair HL and the result is stored in HL. Operand ss is specified as follows in the assembled object code.

Register Pair	ss
BC	00
DE	01
HL	10
SP	11

M CYCLES: 3 T STATES: 11(4,4,3) 4 MHz E.T.: 2.75

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Set if carry out of Bit 11; reset otherwise
P/V: Not affected
N: Reset
C: Set if carry from Bit 15; reset otherwise

Example:

If register pair HL contains the integer 4242H and register pair DE contains 1111H, after the execution of

```
ADD HL,DE
```

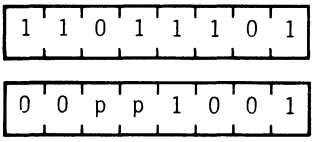
the HL register pair will contain 5353H.

ADD IX,pp

Operation: IX←IX + pp

Format:

Opcode	Operands
ADD	IX,pp



Description:

The contents of register pair pp (any of register pairs BC,DE,IX or SP) are added to the contents of the Index Register IX, and the results are stored in IX. Operand pp is specified as follows in the assembled object code.

Register Pair	pp
BC	00
DE	01
IX	10
SP	11

M CYCLES: 4 T STATES: 15(4,4,4,3) 4 MHz E.T.: 3.75

Condition Bits Affected:

S:	Not affected
Z:	Not affected
H:	Set if carry out of Bit 11; reset otherwise
P/V:	Not affected
N:	Reset
C:	Set if carry from Bit 15; reset otherwise

Example:

If the contents of Index Register IX are 3333H and the contents of register pair BC are 5555H, after the execution of

ADD IX,BC

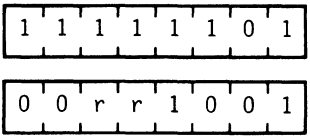
the contents of IX will be 8888H.

ADD IY,rr

Operation: IY←IY + rr

Format:

Opcode	Operands
ADD	IY,rr



Description:

The contents of register pair rr (any of register pairs BC,DE,IY or SP) are added to the contents of Index Register IY, and the result is stored in IY. Operand rr is specified as follows in the assembled object code.

Register Pair	rr
BC	00
DE	01
IY	10
SP	11

M CYCLES: 4 T STATES: 15(4,4,4,3) 4 MHz E.T.: 3.75

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Set if carry out of Bit 11; reset otherwise
P/V: Not affected
N: Reset
C: Set if carry from Bit 15; reset otherwise

Example:

If the contents of Index Register IY are 3333H and the contents of register pair BC are 5555H, after the execution of

ADD IY, BC

the contents of IY will be 8888H.

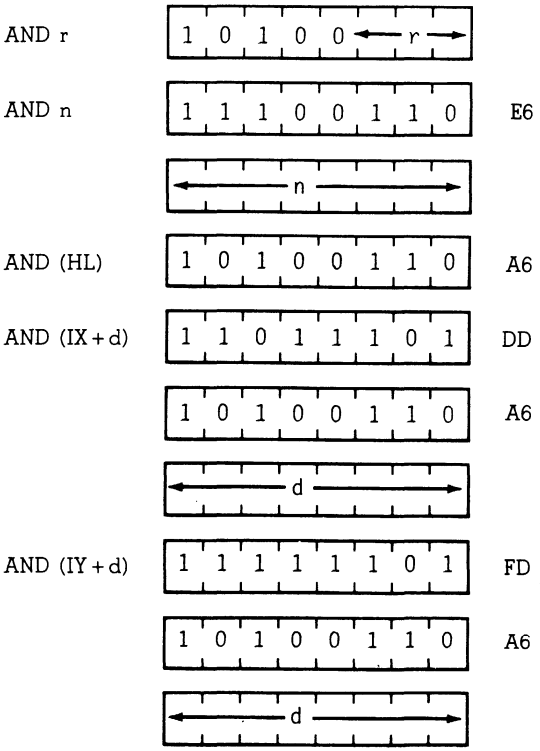
AND s

Operation: A←A∧s

Format:

Opcode	Operands
AND	s

The s operand is any of r,n,(HL),(IX+d) or (IY+d), as defined for the analogous ADD instructions. These various possible opcode-operand combinations are assembled as follows in the object code:



r identifies registers B,C,D,E,H,L or A assembled as follows in the object code field above:

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

A logical AND operation, bit by bit, is performed between the byte specified by the s operand and the byte contained in the Accumulator; the result is stored in the Accumulator.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
AND r	1	4	1.00
AND n	2	7(4,3)	1.75
AND (HL)	2	7(4,3)	1.75
AND (IX+d)	5	19(4,4,3,5,3)	4.75
AND (IX+d)	5	19(4,4,3,5,3)	4.75

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Set
P/V: Set if parity even; reset otherwise
N: Reset
C: Reset

Example:

If the B register contains 7BH (01111011) and the Accumulator contains C3H (11000011) after the execution of

AND B

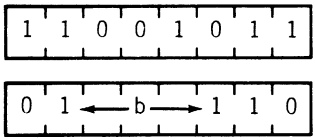
the Accumulator will contain 43H (01000011).

BIT b,(HL)

Operation: $Z \leftarrow \overline{(HL)}_b$

Format:

Opcode	Operands
BIT	b,(HL)



Description:

After the execution of this instruction, the Z flag in the F register will contain the complement of the indicated bit within the contents of the HL register pair. Operand b is specified as follows in the assembled object code:

Bit Tested	b
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

M CYCLES: 3 T STATES: 12(4,4,4) 4 MHz E.T.: 3.00

Condition Bits Affected:

S: Unknown
Z: Set if specified Bit is 0; reset otherwise
H: Set
P/V: Unknown
N: Reset
C: Not affected

Example:

If the HL register pair contains 4444H, and bit 4 in the memory location 4444H contains 1, after the execution of

BIT 4, (HL)

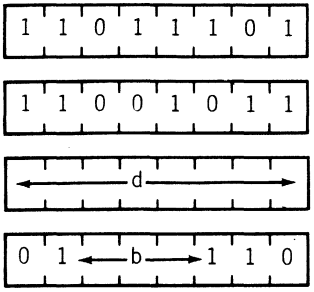
the Z flag in the F register will contain 0, and bit 4 in memory location 4444H will still contain 1.

BIT b,(IX+d)

Operation: $Z \leftarrow \overline{(IX+d)}_b$

Format:

Opcode	Operands
BIT	b,(IX + d)



Description:

After the execution of this instruction, the Z flag in the F register will contain the complement of the indicated bit within the contents of the memory location pointed to by the sum of the contents register pair IX (Index Register IX) and the two's complement displacement integer d. Operand b is specified as follows in the assembled object code.

Bit Tested	b
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

M CYCLES: 5 T STATES: 20(4,4,3,5,4) 4 MHz E.T.: 5.

Condition Bits Affected:

S: Unknown
Z: Set if specified Bit is 0; reset otherwise
H: Set
P/V: Unknown
N: Reset
C: Not affected

BIT b,(IX+d)

Example:

If the contents of Index Register IX are 2000H, and bit 6 in memory location 2004H contains 1, after the execution of

```
BIT 6, (IX+4H)
```

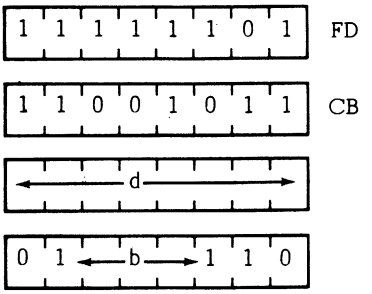
the Z flag in the F register will contain 0, and bit 6 in memory location 2004H will still contain 1.

BIT b,(IY+d)

Operation: $Z \leftarrow \overline{(IY+d)}_b$

Format:

Opcode	Operands
BIT	b,(IY + d)



Description:

After the execution of this instruction, the Z flag in the F register will contain the complement of the indicated bit within the contents of the memory location pointed to by the sum of the contents of register pair IY (Index Register IY) and the two's complement displacement integer d. Operand b is specified as follows in the assembled object code:

Bit Tested	b
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

M CYCLES: 5 T STATES: 20(4,4,3,5,4) 4 MHz E.T.: 5.00

Condition Bits Affected:

S: Unknown
Z: Set if specified Bit is 0; reset otherwise
H: Set
P/V: Unknown
N: Reset
C: Not affected

BIT b,(IY+d)

Example:

If the contents of Index Register IY are 2000H, and bit 6 in memory location 2004H contains 1, after the execution of

```
BIT 6, (IY + 4H)
```

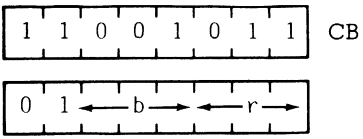
the Z flag in the F register still contain 0, and bit 6 in memory location 2004H will still contain 1.

BIT b,r

Operation: $Z \leftarrow \bar{r}_b$

Format:

Opcode	Operands
BIT	b,r



Description:

After the execution of this instruction, the Z flag in the F register will contain the complement of the indicated bit within the indicated register. Operands b and r are specified as follows in the assembled object code:

Bit tested	b	Register	r
0	000	B	000
1	001	C	001
2	010	D	010
3	011	E	011
4	100	H	100
5	101	L	101
6	110	A	111
7	111		

M CYCLES: 2 T STATES: 8(4,4) 4 MHz E.T.: 2.00

Condition Bits Affected:

S: Unknown
Z: Set if specified Bit is 0; reset otherwise
H: Set
P/V: Unknown
N: Reset
C: Not affected

Example:

If bit 2 in register B contains 0, after the execution of

```
BIT 2,B
```

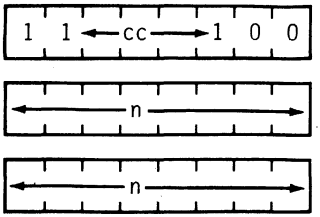
the Z flag in the F register will contain 1, and bit 2 in register B will remain 0.

CALL cc,nn

Operation: IF cc TRUE: $(SP-1) \leftarrow PC_H$
$(SP-2) \leftarrow PC_L$, $PC \leftarrow nn$

Format:

Opcode	Operands
CALL	cc,nn



Note: The first of the two n operands in the assembled object code above is the least significant byte of the two-byte memory address.

Description:

If condition cc is true, this instruction pushes the current contents of the Program Counter (PC) onto the top of the external memory stack, then loads the operands nn into PC to point to the address in memory where the first opcode of a subroutine is to be fetched. (At the end of the subroutine, a RETurn instruction can be used to return to the original program flow by popping the top of the stack back into PC.) If condition cc is false, the Program Counter is incremented as usual, and the program continues with the next sequential instruction. The stack push is accomplished by first decrementing the current contents of the Stack Pointer (SP), loading the high-order byte of the PC contents into the memory address now pointed to by SP; then decrementing SP again, and loading the low-order byte of the PC contents into the top of the stack. Note: Because this is a 3-byte instruction, the Program Counter will have been incremented by 3 before the push is executed. Condition cc is programmed as one of eight status which corresponds to condition bits in the Flag Register (register F). These eight status are defined in the table below, which also specifies the corresponding cc bit fields in the assembled object code:

cc	Condition	Relevant Flag
000	NZ non zero	Z
001	Z zero	Z
010	NC non carry	C
011	C carry	C
100	PO parity odd	P/V
101	PE parity even	P/V
110	P sign positive	S
111	M sign negative	S

If cc is true:

M CYCLES: 5 T STATES: 17(4,3,4,3,3) 4 MHz E.T.: 4.25

If cc is false:

M CYCLES: 3 T STATES: 10(4,3,3) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

If the C Flag in the F register is reset, the contents of the Program Counter are 1A47H, the contents of the Stack Pointer are 3002H, and memory locations have the contents:

Location	**Contents**
1A47H	D4H
1A48H	35H
1A49H	21H

then if an instruction fetch sequence begins, the three-byte instruction D43521H will be fetched to the CPU for execution. The mnemonic equivalent of this is

CALL NC,2135H

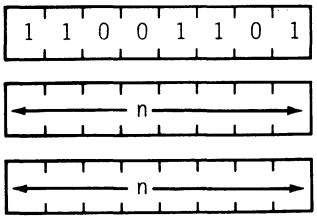
After the execution of this instruction, the contents of memory address 3001H will be 1AH, the contents of address 3000H will be 4AH, the contents of the Stack Pointer will be 3000H, and the contents of the Program Counter will be 2135H, pointing to the address of the first opcode of the subroutine now to be executed.

CALL nn

Operation: (SP-1)←$PC_{H'}$ (SP-2)←$PC_{L'}$ PC←nn

Format:

Opcode	Operands
CALL	nn



Note: The first of the two n operands in the assembled object code above is the least significant byte of a two-byte memory address.

Description:

After pushing the current contents of the Program Counter (PC) onto the top of the external memory stack, the operands nn are loaded into PC to point to the address in memory where the first opcode of a subroutine is to be fetched. (At the end of the subroutine, a RETurn instruction can be used to return to the original program flow by popping the top of the stack back into PC.) The push is accomplished by first decrementing the current contents of the Stack Pointer (register pair SP), loading the high-order byte of the PC contents into the memory address now pointed to by the SP; then decrementing SP again, and loading the low-order byte of the PC contents into the top of stack. Note: Because this is a 3-byte instruction, the Program Counter will have been incremented by 3 before the push is executed.

M CYCLES: 5 T STATES: 17(4,3,4,3,3) 4 MHz E.T.: 4.2

Condition Bits Affected: None

Example:

If the contents of the Program Counter are 1A47H, the contents of the Stack Pointer are 3002H, and memory locations have the contents:

Location	Contents
1A47H	CDH
1A48H	35H
1A49H	21H

then if an instruction fetch sequence begins, the three-byte instruction CD3521H will be fetched to the CPU for execution. The mnemonic equivalent of this is

CALL 2135H

After the execution of this instruction, the contents of memory address 3001H will be 1AH, the contents of address 3000H will be 4AH, the contents of the Stack Pointer will be 3000H, and the contents of the Program Counter will be 2135H, pointing to the address of the first opcode of the subroutine now to be executed.

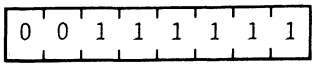
CCF

Operation: CY←$\overline{CY}$

Format:

Opcode

CCF



3F

Description:

The C flag in the F register is inverted.

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected:

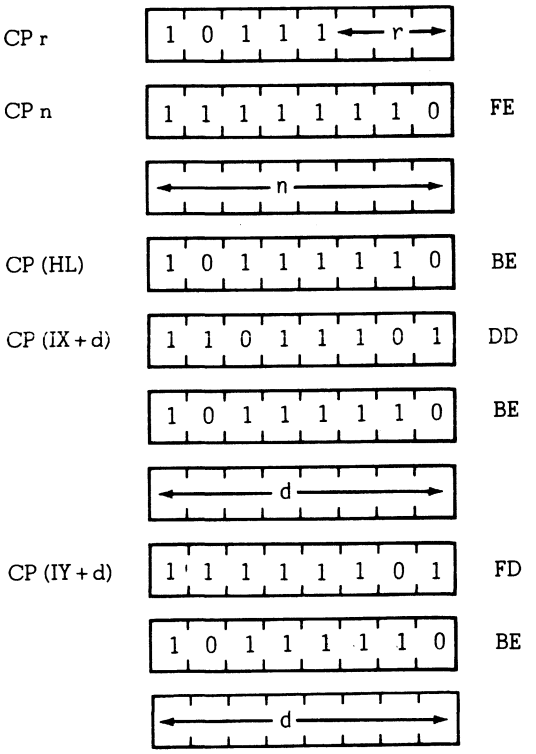
S: Not affected
Z: Not affected
H: Previous carry will be copied
P/V: Not affected
N: Reset
C: Set if CY was 0 before operation; reset otherwise

Operation: A-s

Format:

Opcode	Operands
CP	s

The s operand is any of r,n(HL),(IX + d) or (IY + d), as defined for the analogous ADD instructions. These various possible opcode-operand combinations are assembled as follows in the object code:



r identifies registers B,C,D,E,H,L or A assembled as follows in the object code field above:

CP s

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

The contents of the s operand are compared with the contents of the Accumulator. If there is a true compare, a flag is set.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
CP r	1	4	1.00
CP n	2	7(4,3)	1.75
CP (HL)	2	7(4,3)	1.75
CP (IX + d)	5	19(4,4,3,5,3)	4.75
CP (IY + d)	5	19(4,4,3,5,3)	4.75

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Set in there is a borrow and reset otherwise.
P/V: Set if overflow; reset otherwise
N: Set
C: Set if there is a borrow and reset otherwise.

Example:

If the Accumulator contains 63H, the HL register pair contains 6000H and memory location 6000H contains 60H, the instruction

CP (HL)

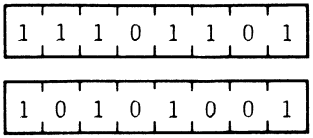
will result in the P/V flag in the F register being reset.

Operation: A-(HL), HL←HL-1, BC←BC-1

Format:

Opcode

CPD



Description:

The contents of the memory location addressed by the HL register pair is compared with the contents of the Accumulator. In case of a true compare, a condition bit is set. The HL and the Byte Counter (register pair BC) are decremented.

M CYCLES: 4 T STATES: 16(4,4,3,5) 4 MHz E.T.: 4.00

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if A = (HL); reset otherwise
H:	Set if there is a borrow and reset otherwise.
P/V:	Set if BC-1 ≠ 0; reset otherwise
N:	Set
C:	Not Affected

Example:

If the HL register pair contains 1111H, memory location 1111H contains 3BH, the Accumulator contains 3BH, and the Byte Counter contains 0001H, then after the execution of

CPD

the Byte Counter will contain 0000H, the HL register pair will contain 1110H, the Z flag in the F register will be set, and the P/V flag in the F register will be reset. There will be no effect on the contents of the Accumulator or address 1111H.

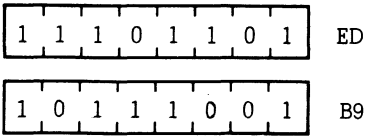
CPDR

Operation: A - (HL), HL←HL-1, BC←BC-1

Format:

Opcode

CPDR



Description:

The contents of the memory location addressed by the HL register pair is compared with the contents of the Accumulator. In case of a true compare, a condition bit is set. The HL and BC (Byte Counter) register pairs are decremented. If decrementing causes the BC to go to zero or if A = (HL), the instruction is terminated. If BC is not zero and A≠(HL), the program counter is decremented by 2 and the instruction is repeated. Note that if BC is set to zero prior to instruction execution, the instruction will loop through 64K bytes, if no match is found. Interrupts will be recognized and two refresh cycles will be executed after each data transfer.

For BC≠0 and A≠(HL):

M CYCLES: 5 T STATES: 21(4,4,3,5,5) 4 MHz E.T.: 5.25

For BC = 0 or A = (HL):

M CYCLES: 4 T STATES: 16(4,4,3,5) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if A = (HL); reset otherwise
H: Set if there is a borrow and reset otherwise.
P/V: Set if BC-1≠0; reset otherwise
N: Set
C: Not affected

Example:

If the HL register pair contains 1118H, the Accumulator contains F3H, the Byte Counter contains 0007H, and memory locations have these contents:

(1118H) : 52H
(1117H) : 00H
(1116H) : F3H

then after the execution of

CPDR

the contents of register pair HL will be 1115H, the contents of the Byte Counter will be 0004H, the P/V flag in the F register will be set, and the Z flag in the F register will be set.

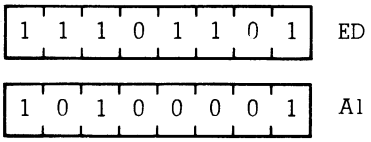
CPI

Operation: A - (HL), HL←HL+1, BC←BC-1

Format:

Opcode

CPI



Description:

The contents of the memory location addressed by the HL register pair is compared with the contents of the Accumulator. In case of a true compare, a condition bit is set. Then HL is incremented and the Byte Counter (register pair BC) is decremented.

M CYCLES: 4 T STATES: 16(4,4,3,5) 4 MHz E.T.: 4.00

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if A = (HL); reset otherwise
H:	Set if there is a borrow and reset otherwise.
P/V:	Set if BC-1 ≠ 0; reset otherwise
N:	Set
C:	Not affected

Example:

If the HL register pair contains 1111H, memory location 1111H contains 3BH, the Accumulator contains 3BH, and the Byte Counter contains 0001H, then after the execution of

CPI

the Byte Counter will contain 0000H, the HL register pair will contain 1112H, the Z flag in the F register will be set, and the P/V flag in the F register will be reset. There will be no effect on the contents of the Accumulator or address 1111H.

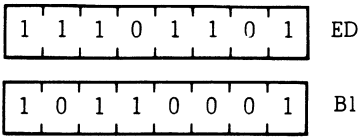
CPIR

Operation: A - (HL), HL←HL+1, BC←BC-1

Format:

Opcode

CPIR



Description:

The contents of the memory location addressed by the HL register pair is compared with the contents of the Accumulator. In case of a true compare, a condition bit is set. The HL is incremented and the Byte Counter (register pair BC) is decremented. If decrementing causes the BC to go to zero or if A = (HL), the instruction is terminated. If BC is not zero and A≠(HL), the program counter is decremented by 2 and the instruction is repeated. Note that if BC is set to zero before instruction execution, the instruction will loop through 64K bytes, if no match is found. Interrupts will be recognized and two refresh cycles will be executed after each data transfer.

For BC≠0 and A≠(HL):

M CYCLES: 5 T STATES: 21(4,4,3,5,5) 4 MHz E.T.: 5.25

For BC = 0 or A = (HL):

M CYCLES: 4 T STATES: 16(4,4,3,5) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if A = (HL); reset otherwise
H: Set if there is a borrow and reset otherwise.
P/V: Set if BC-1≠0; reset otherwise
N: Set
C: Not affected

CPIR

Example:

If the HL register pair contains 1111H, the Accumulator contains F3H, the Byte Counter contains 0007H, and memory locations have these contents:

(1111H) : 52H
(1112H) : 00H
(1113H) : F3H

then after the execution of

CPIR

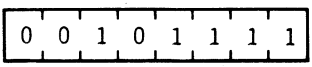
the contents of register pair HL will be 1114H, the contents of the Byte Counter will be 0004H, the P/V flag in the F register will be set and the Z flag in the F register will be set.

Operation: $A \leftarrow \overline{A}$

Format:

Opcode

CPL



2F

Description:

Contents of the Accumulator (register A) are inverted (1's complement).

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Set
P/V: Not affected
N: Set
C: Not affected

Example:

If the contents of the Accumulator are 1011 0100, after the execution of

CPL

the Accumulator contents will be 0100 1011.

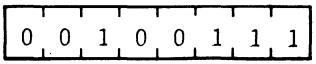
DAA

Operation: Decimal Adjust Accumulator

Format:

Opcode

DAA



27

Description:

This instruction conditionally adjusts the Accumulator for BCD addition and subtraction operations. For addition (ADD, ADC, INC) or subtraction (SUB, SBC, DEC, NEG), the following table indicates operation performed:

OPERATION	C BEFORE DAA	HEX VALUE IN UPPER DIGIT (bit 7-4)	H BEFORE DAA	HEX VALUE IN LOWER DIGIT (bit 3-0)	NUMBER ADDED TO BYTE	C AFTER DAA
	0	0-9	0	0-9	00	0
	0	0-8	0	A-F	06	0
	0	0-9	1	0-3	06	0
ADD	0	A-F	0	0-9	60	1
ADC	0	9-F	0	A-F	66	1
INC	0	A-F	1	0-3	66	1
	1	0-2	0	0-9	60	1
	1	0-2	0	A-F	66	1
	1	0-3	1	0-3	66	1
SUB	0	0-9	0	0-9	00	0
SBC	0	0-8	1	6-F	FA	0
DEC	1	7-F	0	0-9	A0	1
NEG	1	6-F	1	6-F	9A	1

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected:

S:	Set if most significant bit of Acc. is 1 after operation; reset otherwise
Z:	Set if Acc. is zero after operation; reset otherwise
H:	See instruction
P/V:	Set if Acc. is even parity after operation; reset otherwise
N:	Not affected
C:	See instruction

Example:

If an addition operation is performed between 15 (BCD) and 27 (BCD), simple decimal arithmetic gives this result:

```
  15
+ 27
----
  42
```

But when the binary representations are added in the Accumulator according to standard binary arithmetic,

```
  0001  0101
+ 0010  0111
-----------
  0011  1100   3C
```

the sum is ambiguous. The DAA instruction adjusts this result so that the correct BCD representation is obtained:

```
  0011  1100
+ 0000  0110
-----------
  0100  0010  = 42
```

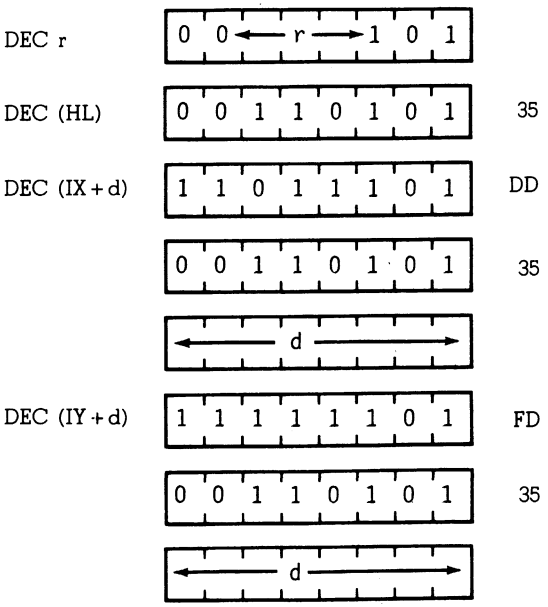
DEC m

Operation: m←m-1

Format:

Opcode	Operands
DEC	m

The m operand is any of r, (HL), (IX + d) or (IY + d), as defined for the analogous INC instructions. These various possible opcode-operand combinations are assembled as follows in the object code:



r identifies registers B,C,D,E,H,L or A assembled as follows in the object code field above:

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

The byte specified by the m operand is decremented.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
DEC r	1	4	1.00
DEC (HL)	3	11(4,4,3)	2.75
DEC (IX+d)	6	23(4,4,3,5,4,3)	5.75
DEC (IY+d)	6	23(4,4,3,5,4,3)	5.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if there is a borrow and reset otherwise.
P/V:	Set if m was 80H before operation; reset otherwise
N:	Set
C:	Not affected

Example:

If the D register contains byte 2AH, after the execution of

DEC D

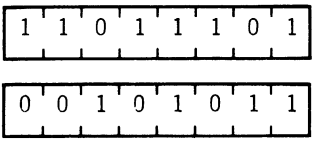
register D will contain 29H.

DEC IX

Operation: IX←IX-1

Format:

Opcode	Operands
DEC	IX



Description:

The contents of Index Register IX are decremented.

M CYCLES: 2 T STATES: 10(4,6) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

If the contents of Index Register IX are 2006H, after the execution of

```
DEC IX
```

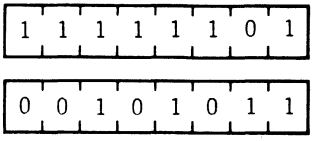
the contents of Index Register IX will be 2005H.

DEC IY

Operation: IY←IY-1

Format:

Opcode	Operands
DEC	IY



Description:

The contents of the Index Register IY are decremented.

M CYCLES: 2 T STATES: 10 (4,6) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

If the contents of the Index Register IY are 7649H, after the execution of

```
DEC IY
```

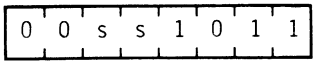
the contents of Index Register IY will be 7648H.

DEC ss

Operation: ss←ss-1

Format:

Opcode	Operands
DEC	ss



Description:

The contents of register pair ss (any of the register pairs BC,DE,HL or SP) are decremented. Operand ss is specified as follows in the assembled object code.

Pair	ss
BC	00
DE	01
HL	10
SP	11

M CYCLES: 1 T STATES: 6 4 MHz E.T.: 1.50

Condition Bits Affected: None

Example:

If register pair HL contains 1001H, after the execution of

```
DEC HL
```

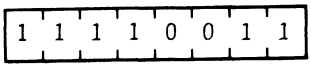
the contents of HL will be 1000H.

Operation: IFF←0

Format:

Opcode

DI



F3

Description:

DI disables the maskable interrupt by resetting the interrupt enable flip-flops (IFF1 and IFF2). Note that this instruction disables the maskable interrupt during its execution.

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected: None

Example:

When the CPU executes the instruction

DI

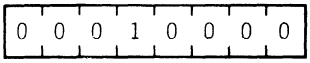
the maskable interrupt is disabled until it is subsequently re-enabled by an EI instruction. The CPU will not respond to an Interrupt Request (INT) signal.

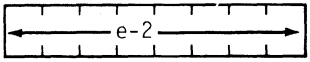
DJNZ e

Operation: Jump relative and decrement.

Format:

Opcode	Operand
DJNZ	e





Description:

This instruction is similar to the conditional jump instructions except that a register value is used to determine branching. The B register is decremented and if a non zero value remains, the value of the displacement e is added to the Program Counter (PC). The next instruction is fetched from the location designated by the new contents of the PC. The jump is measured from the address of the instruction opcode and has a range of -126 to + 129 bytes. The assembler calculates the displacement e and automatically adjusts for the twice incremented PC. If the result of decrementing leaves B with a zero value, the next instruction to be executed is taken from the location following this instruction.

If $B \neq 0$;

M CYCLES: 3 T STATES: 13(5,3,5) 4 MHz E.T.: 3.25

If $B \neq 0$:

M CYCLES: 2 T STATES: 8(5,3) 4 MHz E.T.: 2.00

Condition Bits Affected: None

Example:

A typical software routine is used to demonstrate the use of the DJNZ instruction. This routine moves a line from an input buffer (INBUF) to an output buffer (OUTBUF). It moves the bytes until it finds a CR, or until it has moved 80 bytes, whichever occurs first.

```
        LD      B,80        ; Set up counter
        LD      HL,INBUF    ; Set up pointers
        LD      DE,OUTBUF

LOOP:   LD      A,(HL)      ; Get next byte from
                            ; input buffer
        LD      (DE),A      ; Store in output buffer
        CP      0DH         ; Is it a CR?
        JR      Z,DONE      ; Yes finished
        INC     HL          ; Increment pointers
        INC     DE
        DJNZ    LOOP        ; Loop back if 80
                            ; bytes have not
                            ; been moved
DONE:
```

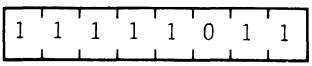
EI

Operation: IFF←1

Format:

Opcode

EI



Description:

EI enables the maskable interrupt by setting the interrupt enable flip-flops (IFF1 and IFF2). Note that this instruction disables the maskable interrupt during its execution.

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected: None

Example:

When the CPU executes instruction

EI

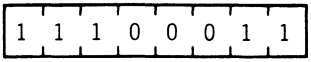
the maskable interrupt is enabled. The CPU will now respond to an Interrupt Request (INT) signal.

EX (SP),HL

Operation: H↔(SP + 1), L↔(SP)

Format:

Opcode	Operands
EX	(SP),HL



E3

Description:

The low order byte contained in register pair HL is exchanged with the contents of the memory address specified by the contents of register pair SP (Stack Pointer), and the high order byte of HL is exchanged with the next highest memory address (SP + 1).

M CYCLES: 5 T STATES: 19(4,3,4,3,5) 4 MHz E.T.: 4.75

Condition Bits Affected: None

Example:

If the HL register pair contains 7012H, the SP register pair contains 8856H, the memory location 8856H contains the byte 11H, and the memory location 8857H contains the byte 22H, then the instruction

EX (SP), HL

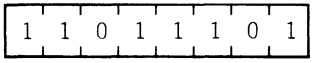
will result in the HL register pair containing number 2211H, memory location 8856H containing the byte 12H, the memory location 8857H containing the byte 70H and the Stack Pointer containing 8856H.

EX (SP),IX

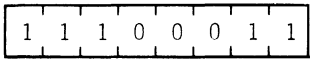
Operation: $IX_H \leftrightarrow (SP+1)$, $IX_L \leftrightarrow (SP)$

Format:

Opcode	Operands
EX	(SP),IX



DD



E3

Description:

The low order byte in Index Register IX is exchanged with the contents of the memory address specified by the contents of register pair SP (Stack Pointer), and the high order byte of IX is exchanged with the next highest memory address (SP + 1).

M CYCLES: 6 T STATES: 23(4,4,3,4,3,5) 4 MHz E.T.: 5.75

Condition Bits Affected: None

Example:

If the Index Register IX contains 3988H, the SP register pair contains 0100H, the memory location 0100H contains the byte 90H, and memory location 0101H contains byte 48H, then the instruction

```
EX (SP),IX
```

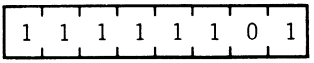
will result in the IX register pair containing number 4890H, memory location 0100H containing 88H, memory location 0101H containing 39H and the Stack Pointer containing 0100H.

EX (SP),IY

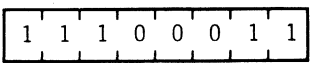
Operation: $IY_H \leftrightarrow (SP+1)$, $IY_L \leftrightarrow (SP)$

Format:

Opcode	Operands
EX	(SP),IY



FD



E3

Description:

The low order byte in Index Register IY is exchanged with the contents of the memory address specified by the contents of register pair SP (Stack Pointer), and the high order of IY is exchanged with the next highest memory address (SP + 1).

M CYCLES: 6 T STATES: 23(4,4,3,4,3,5) 4 MHz E.T.: 5.75

Condition Bits Affected: None

Example:

If the Index Register IY Contains 3988H, the SP register pair contains 0100H, the memory location 0100H contains the byte 90H, and memory location 0101H contains byte 48H, then the instruction

EX (SP), IY

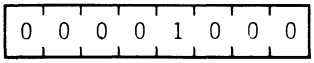
will result in the IY register pair containing number 4890H, memory location 0100H containing 88H, memory location 0101H containing 39H, and the Stack Pointer containing 0100H.

EX AF,AF'

Operation: AF↔AF'

Format:

Opcode	Operands
EX	AF,AF'



08

Description:

The two-byte contents of the register pairs AF and AF' are exchanged. (Note: register pair AF' consists of registers A' and F'.)

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected: None

Example:

If the content of register pair AF is number 9900H, and the content of register pair AF' is number 5944H, after the instruction

```
EX AF,AF'
```

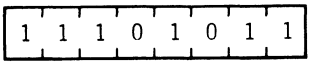
the contents of AF will be 5944H, and the contents of AF' will be 9900H.

EX DE,HL

Operation: DE↔HL

Format:

Opcode	Operands
EX	DE,HL



EB

Description:

The two-byte contents of register pairs DE and HL are exchanged.

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected: None

Example:

If the content of register pair DE is the number 2822H, and the content of the register pair HL is number 499AH, after the instruction

```
EX DE,HL
```

the content of register pair DE will be 499AH and the content of register pair HL will be 2822H.

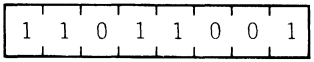
EXX

Operation: (BC)↔(BC'), (DE)↔(DE'), (HL)↔(HL')

Format:

Opcode

EXX



D9

Description:

Each two-byte value in register pairs BC, DE, and HL is exchanged with the two-byte value in BC', DE', and HL', respectively.

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected: None

Example:

If the contents of register pairs BC, DE, and HL are the numbers 445AH, 3DA2H, and 8859H, respectively, and the contents of register pairs BC', DE', and HL' are 0988H, 9300H, and 00E7H, respectively, after the instruction

```
EXX
```

the contents of the register pairs will be as follows: BC: 0988H; DE: 9300H; HL: 00E7H; BC': 445AH; DE': 3DA2H; and HL': 8859H.

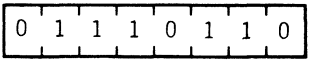
HALT

Operation: CPU Halt

Format:

Opcode

HALT



76

Description:

The HALT instruction suspends CPU operation until a subsequent interrupt or reset is received. While in the halt state, the processor will execute NOP's to maintain memory refresh logic.

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

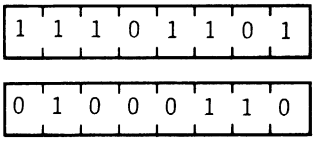
Condition Bits Affected: None

IM 0

Operation: Interrupt Mode 0

Format:

Opcode	Operands
IM	0



Description:

The IM 0 instruction sets interrupt mode 0. In this mode the interrupting device can insert any instruction on the data bus and allow the CPU to execute it.

M CYCLES: 2 T STATES: 8(4,4) 4 MHz E.T.: 2.00

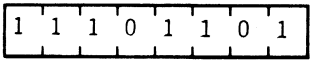
Condition Bits Affected: None

IM 1

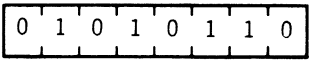
Operation: Interrupt Mode 1

Format:

Opcode	Operands
IM	1



ED



56

Description:

The IM instruction sets interrupt mode 1. In this mode the processor will respond to an interrupt by executing a restart to location 0038H.

M CYCLES: 2 T STATES: 8(4,4) 4 MHz E.T.: 2.00

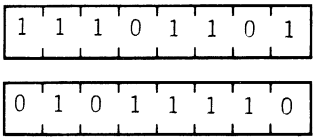
Condition Bits Affected: None

IM 2

Operation: Interrupt Mode 2.

Format:

Opcode	Operands
IM	2



Description:

The IM 2 instruction sets interrupt mode 2. This mode allows an indirect call to any location in memory. With this mode the CPU forms a 16-bit memory address. The upper eight bits are the contents of the Interrupt Vector Register I and the lower eight bits are supplied by the interrupting device.

M CYCLES: 2 T STATES: 8(4,4) 4 MHz E.T.: 2.00

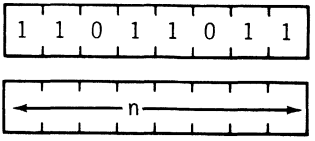
Condition Bits Affected: None

IN A,(n)

Operation: A←(n)

Format:

Opcode	Operands
IN	A,(n)



Description:

The operand n is placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. The contents of the Accumulator also appear on the top half (A8 through A15) of the address bus at this time. Then one byte from the selected port is placed on the data bus and written into the Accumulator (register A) in the CPU.

M CYCLES: 3 T STATES: 11(4,3,4) 4 MHz E.T.: 2.75

Condition Bits Affected: None

Example:

If the contents of the Accumulator are 23H and the byte 7BH is available at the peripheral device mapped to I/O port address 01H, then after the execution of

```
IN A,(01H)
```

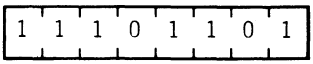
the Accumulator will contain 7BH.

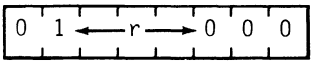
IN r,(C)

Operation: r←(C)

Format:

Opcode	Operands
IN	r,(C)





Description:

The contents of regsiter C are placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. The contents of Register B are placed on the top half (A8 through A15) of the address bus at this time. Then one byte from the selected port is placed on the data bus and written into register r in the CPU. Register r identifies any of the CPU registers shown in the following table, which also shows the corresponding 3-bit r field for each. The flags will be affected, checking the input data.

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

M CYCLES: 3 T STATES: (12(4,4,4) 4 MHz E.T.: 3.00

Condition Bits Affected:

S:	Set if input data is negative; reser otherwise
Z:	Set if input data is zero; reset otherwise
H:	Reset
P/V:	Set if parity is even; reset otherwise
N:	Reset
C:	Not affected

Example:

If the contents of register C are 07H, the contents of register B are 10H, and the byte 7BH is available at the peripheral device mapped to I/O port address 07H, then after the execution of

IN D,(C)

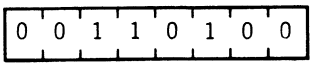
The register D will contain 7BH.

INC (HL)

Operation: (HL)←(HL) + 1

Format:

Opcode	Operands
INC	(HL)



34

Description:

The byte contained in the address specified by the contents of the HL register pair is incremented.

M CYCLES: 3 T STATES: 11(4,4,3) 4 MHz E.T.: 2.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if carry from Bit 3; reset otherwise
P/V:	Set if (HL) was 7FH before operation; reset otherwise
N:	Reset
C:	Not Affected

Example:

If the contents of the HL register pair are 3434H, and the contents of address 3434H are 82H, after the execution of

INC (HL)

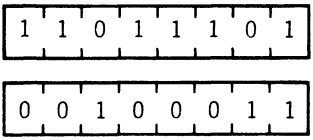
memory location 3434H will contain 83H.

INC IX

Operation: IX←IX + 1

Format:

Opcode	Operands
INC	IX



Description:

The contents of the Index Register IX are incremented.

M CYCLES: 2 T STATES: 10(4,6) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

If the Index Register IX contains the integer 3300H after the execution of

```
INC IX
```

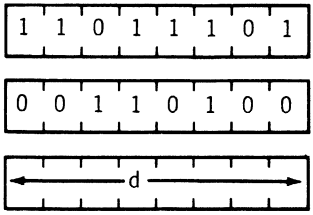
the contents of Index Register IX will be 3301H.

INC (IX+d)

Operation: (IX + d)←(IX + d) + 1

Format:

Opcode	Operands
INC	(IX + d)



Description:

The contents of the Index Register IX (register pair (IX) are added to a two's complement displacement integer d to point to an address in memory. The contents of this address are then incremented.

M CYCLES: 6 T STATES: 23(4,4,3,5,4,3) 4 MHz E.T.: 5.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if carry from Bit 3; reset otherwise
P/V:	Set if (IX + d) was 7FH before operation; reset otherwise
N:	Reset
C:	Not affected

Example:

If the contents of the Index Register pair IX are 2020H, and the memory location 2030H contains byte 34H, after the execution of

INC ((IX + 10H)

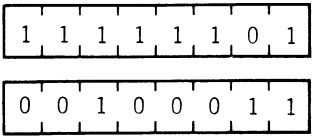
the contents of memory location 2030H will be 35H.

INC IY

Operation: IY←IY + 1

Format:

Opcode	Operands
INC	IY



Description:

The contents of the Index Register IY are incremented.

M CYCLES: 2 T STATES: 10(4,6) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

If the contents of the Index Register are 2977H, after the execution of

```
INC IY
```

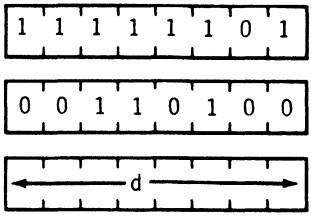
the contents of Index Register IY will be 2978H.

INC (IY+d)

Operation: (IY + d)←(IY + d) + 1

Format:

Opcode	Operands
INC	(IY + d)



Description:

The contents of the Index Register IY (register pair IY) are added to a two's complement displacement integer d to point to an address in memory. The contents of this address are then incremented.

M CYCLES: 6 T STATES: 23(4,4,3,5,4,3) 4 MHz E.T.: 5.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if carry from Bit 3; reset otherwise
P/V:	Set if (IY + d)was 7FH before operation; reset otherwise
N:	Reset
C:	Not Affected

Example:

If the contents of the Index Register pair IY are 2020H, and the memory location 2030H contains byte 34H, after the execution of

INC (IY + 10H)

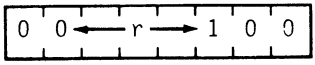
the contents of memory location 2030H will be 35H.

INC r

Operation: r←r + 1

Format:

Opcode	Operands
INC	r



Description:

Register r is incremented, r identifies any of the registers A,B,C,D,E,H or L, assembled as follows in the object code.

Register	r
A	111
B	000
C	001
D	010
E	011
H	100
L	101

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if carry from Bit 3; reset otherwise
P/V:	Set if r was 7FH before operation; reset otherwise
N:	Reset
C:	Not affected

Example:

If the contents of register D are 28H, after the execution of

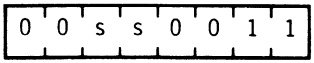
```
INC D
```

the contents of register D will be 29H.

Operation: ss←ss + 1

Format:

Opcodes	Operands
INC	ss



Description:

The contents of register pair ss (any of register pairs BC, DE,HL or SP) are incremented. Operand ss is specified as follows in the assembled object code.

Register Pair	ss
BC	00
DE	01
HL	10
SP	11

M CYCLES: 1 T STATES: 6 4 MHz E.T. 1.50

Condition Bits Affected: None

Example:

If the register pair contains 1000H, after the execution of

```
INC HL
```

HL will contain 1001H.

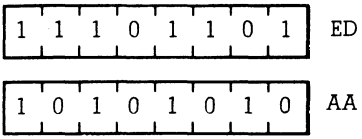
IND

Operation: (HL)←(C), B←B-1, HL←HL-1

Format:

Opcode

IND



Description:

The contents of register C are placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. Register B may be used as a byte counter, and its contents are placed on the top half (A8 through A15) of the address bus at this time. Then one byte from the selected port is placed on the data bus and written to the CPU. The contents of the HL register pair are placed on the address bus and the input byte is written into the corresponding location of memory. Finally the byte counter and register pair HL are decremented.

M CYCLES: 4 T STATES: 16(4,5,3,4) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Unknown
Z: Set if B-1 = 0; reset otherwise
H: Unknown
P/V: Unknown
N: Set
C: Not affected

Example:

If the contents of register C are 07H, the contents of register B are 10H, the contents of the HL register pair are 1000H, and the byte 7BH is available at the peripheral device mapped to I/O port address 07H, then after the execution of

```
IND
```

memory location 1000H will contain 7BH, ,the HL register pair will contain 0FFFH, and register B will contain 0FH.

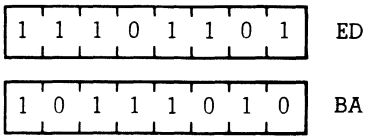
INDR

Operation: (HL)←(C), B←B-1, HL←HL-1

Format:

Opcode

INDR



Description:

The contents of register C are placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. Register B is used as a byte counter, and its contents are placed on the top half (A8 through A15) of the address bus at this time. Then one byte from the selected port is placed on the data bus and written to the CPU. The contents of the HL register pair are placed on the address bus and the input byte is written into the corresponding location of memory. Then HL and the byte counter are decremented. If decrementing causes B to go to zero, the instruction is terminated. If B is not zero, the PC is decremented by two and the instruction repeated. Note that if B is set to zero prior to instruction execution, 256 bytes of data will be input. Interrupts will be recognized and two refresh cycles will be executed after each data transfer.

IF B≠0:

M CYCLES: 5 T STATES: 21(4,5,3,4,5) 4 MHz E.T.: 5.25

If B=0:

M CYCLES: 4 T STATES: 16(4,5,3,4) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Unknown
Z: Set
H: Unknown
P/V: Unknown
N: Set
C: Not affected

INDR

Example:

If the contents of register C are 07H, the contents of register B are 03H, the contents of the HL register pair are 1000H, and the following sequence of bytes are available at the peripheral device mapped to I/O port address 07H:

51H
A9H
03H

then after the execution of

INDR

the HL register pair will contain 0FFDH, register B will contain zero, and memory locations will have contents as follows:

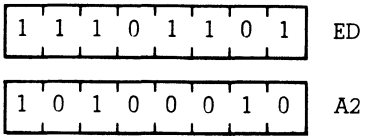
Location	**Contents**
0FFEH	03H
0FFFH	A9H
1000H	51H

Operation: (HL)←(C), B←B-1, HL←HL+1

Format:

Opcode

INI



Description:

The contents of register C are placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. Register B may be used as a byte counter, and its contents are placed on the top half (A8 through A15) of the address bus at this time. Then one byte from the selected port is placed on the data bus and written to the CPU. The contents of the HL register pair are then placed on the address bus and the input byte is written into the corresponding location of memory. Finally the byte counter is decremented and register pair HL is incremented.

M CYCLES: 4 T STATES: 16(4,5,3,4) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Unknown
Z: Set if B-1 = 0; reset otherwise
H: Unknown
P/V: Unknown
N: Set
C: Not affected

Example:

If the contents of register C are 07H, the contents of register B are 10H, the contents of the HL register pair are 1000H, and the byte 7BH is available at the peripheral device mapped to I/O port address 07H, then after the execution of

```
INI
```

memory location 1000H will contain 7BH, the HL register pair will contain 1001H, and register B will contain 0FH.

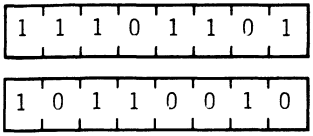
INIR

Operation: (HL)←(C), B←B-1, HL←HL+1

Format:

Opcode

INIR



Description:

The contents of register C are placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. Register B is used as a byte counter, and its contents are placed on the top half (A8 through A15) of the address bus at this time. Then one byte from the selected port is placed on the data bus and written to the CPU. The contents of the HL register pair are placed on the address bus and the input byte is written into the corresponding location of memory. Then register pair HL is incremented, the byte counter is decremented. If decrementing causes B to go to zero, the instruction is terminated. If B is not zero, the PC is decremented by two and the instruction repeated. Note that if B is set to zero prior to instruction execution, 256 bytes of data will be input. Interrupts will be recognized and two refresh cycles will be executed after each data transfer.

If B≠0:

M CYCLES: 5 T STATES: 21(4,5,3,4,5) 4 MHz E.T.: 5.25

If B=0:

M CYCLES: 4 T STATES: 16(4,5,3,4) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Unknown
Z: Set
H: Unknown
P/V: Unknown
N: Set
C: Not affected

Example:

If the contents of register C are 07H, the contents of register B are 03H, the contents of the HL register pair are 1000H, and the following sequence of bytes are available at the peripheral device mapped to I/O port of address 07H:

51H
A9H
03H

the after the execution of

INIR

the HL register pair will contain 1003H, register B will contain zero, and memory locations will have contents as follows:

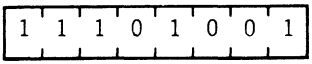
Location	**Contents**
1000H	51H
1001H	A9H
1002H	03H

JP (HL)

Operation: PC←HL

Format:

Opcode	Operands
JP	(HL)



E9

Description:

The Program Counter (register pair PC) is loaded with the contents of the HL register pair. The next instruction is fetched from the location designated by the new contents of the PC.

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected: None

Example:

If the contents of the Program Counter are 1000H and the contents of the HL register pair are 4800H, after the execution of

```
JP (HL)
```

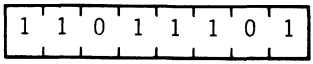
the contents of the Program Counter will be 4800H.

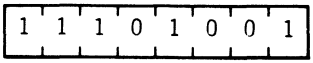
JP (IX)

Operation: PC←IX

Format:

Opcode	Operands
JP	(IX)





Description:

The program Counter (register pair PC) is loaded with the contents of the IX Register Pair (Index Register IX). The next instruction is fetched from the location designated by the new contents of the PC.

M CYCLES: 2 STATES: 8(4,4) 4 MHz E.T.: 2.00

Condition Bits Affected: None

Example:

If the contents of the Program Counter are 1000H, and the contents of the IX Register Pair are 4800H, after the execution of

```
JP (IX)
```

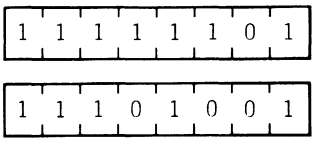
the contents of the Program Counter will be 4800H.

JP (IY)

Operation: PC←IY

Format:

Opcode	Operands
JP	(IY)



Description:

The Program Counter (register pair PC) is loaded with the contents of the IY register pair (Index Register IY). The next instruction is fetched from the location designated by the new contents of the PC.

M CYCLES: 2 T STATES: 8(4,4) 4 MHz E.T.: 2.00

Condition Bits Affected: None

Example:

If the contents of the Program Counter are 1000H and the contents of the IY Register Pair are 4800H, after the execution of

```
JP (IY)
```

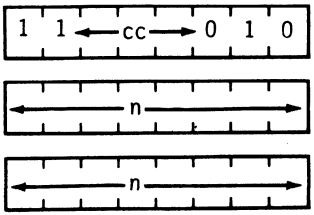
the contents of the Program Counter will be 4800H.

JP cc,nn

Operation: IF cc TRUE, PC←nn

Format:

Opcode	Operands
JP	cc,nn



Note: The first n operand in this assembled object code is the low order byte of a 2-byte memory address.

Description:

If condition cc is true, the instruction loads operand nn into register pair PC (Program Counter), and the program continues with the instruction beginning at address nn. If condition cc is false, the Program Counter is incremented as usual, and the program continues with the next sequential instruction.
Condition cc is programmed as one of eight status which corresponds to condition bits in the Flag Register (register F). These eight status are defined in the table fields in the assembled object code.

cc	CONDITIONS	RELEVANT FLAG
000	NZ non zero	Z
001	Z zero	Z
010	NC no carry	C
011	C carry	C
100	PO parity odd	P/V
101	PE parity even	P/V
110	P sign positive	S
111	M sign negative	S

M CYCLES: 3 T STATES: 10(4,3,3) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

If the Carry Flag (C flag in the F register) is set and the contents of address 1520H are 03H, after the execution of

```
JP C,1520H
```

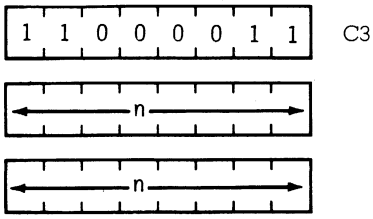
the Program Counter will contain 1520H, and on the next machine cycle the CPU will fetch from address 1520H the byte 03H.

JP nn

Operation: PC←nn

Format:

Opcode	Operands
JP	nn



Note: The first operand in this assembled object code is the low order byte of a 2-byte address.

Description:

Operand nn is loaded into register pair PC (Program Counter) and points to the address of the next program instruction to be executed.

M CYCLES: 3 T STATES: 10(4,3,3) 4 MHz E.T.: 2.50

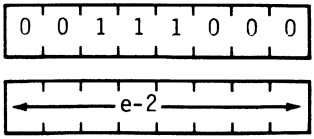
Condition Bits Affected: None

JR C,e

Operation: If C = 0, continue
If C = 1, PC←PC + e

Format:

Opcode	Operands
JR	C,e



Description:

This instruction provides for conditional branching to other segments of a program depending on the results of a test on the Carry Flag. If the flag is equal to a '1', the value of the displacement e is added to the Program Counter (PC) and the next instruction is fetched from the location designated by the new contents of the PC. The jump is measured from the address of the instruction opcode and has a range of -126 to + 129 bytes. The assembler calculates the displacement e and automatically adjusts for the twice incremented PC.

If the flag is equal to '0', the next instruction to be executed is taken from the location following this instruction.

If condution is met:

M CYCLES: 3 T STATES: 12(4,3,5) 4 MHz E.T.: 3.00

If condition is not met:

M CYCLES: 2 T STATES: 7(4,3) 4 MHz E.T.: 1.75

Condition Bits Affected: None

Example:

The Carry Flag is set and it is required to jump back 4 locations from 480H. The assembly language statement is:

```
JR C,LABEL
```

The resulting object code and final PC value is shown below:

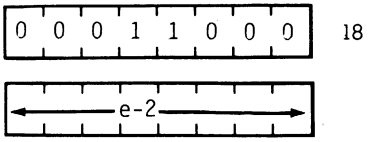
Location				Opcode
47C	LABEL:	—		PC after jump
47D		—		—
47E		—		—
47F		—		—
480		JR	C, LABEL	38
481				FA (2's complement-6)

JR e

Operation: PC←PC + e

Format:

Opcode	Operands
JR	e



Description:

This instruction provides for unconditional branching to other segments of a program. The value of the displacement e is added to the Program Counter (PC) and the next instruction is fetched from the location designated by the new contents of the PC. This jump is measured from the address of the instruction opcode and has a range of -126 to + 129 bytes. The assembler calculates the displacement e and automatically adjusts for the twice incremented PC.

M CYCLES: 3 T STATES: 12(4,3,5) 4 MHz E.T.: 3.00

Condition Bits Affected: None

Example:

To jump forward 5 locations from address 480, the following assembly language statement is used:

```
JR LABEL
```

The resulting object code and final PC value is shown below:

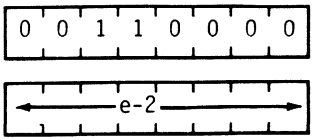
Location				Opcode
480		JR	LABEL	18
481		—		03
482		—		
483		—		
484		—		
485	LABEL:	—		PC after jump

JR NC,e

Operation: If C = 1, continue
If C = 0, PC←PC + e

Format:

Opcode	Operands
JR	NC,e



Description:

This instruction provides for conditional branching to other segments of a program depending on the results of a test on the Carry Flag. If the flag is equal to '0', the value of the displacement e is added to the Program Counter (PC) and the next instruction is fetched from the location designated by the new contents of the PC. The jump is measured from the address of the instruction opcode and has a range of -126 to + 129 bytes. The assembler calculates the displacement e and automatically adjusts for the twice incremented PC.

If the flag is equal to a '1', the next instruction to be executed is taken from the location following this instruction.

If the condition is met:

M CYCLES: 3 T STATES: 12(4,3,5) 4 MHz E.T.: 3.00

If the condition is not met:

M CYCLES: 7 T STATES: 7(4,3) 4 MHz E.T.: 1.75

Condition Bits Affected: None

Example:

The Carry Flag is reset and it is required to repeat the jump instruction. The assembly language statement is:

```
LABEL: JR NC,LABEL
```

The resulting object code and PC after the jump are shown below:

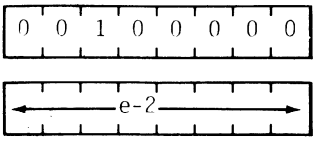
Location				Opcode
480	LABEL:	JR	NC, LABEL	30 ← PC after jump
481				FE (2's complement-2)

JR NZ,e

Operation: If Z = 1, continue
If Z = 0, PC←PC + e

Format:

Opcode	Operands
JR	NZ,e



Description:

This instruction provides for conditional branching to other segments of a program depending on the results of a test on the Zero Flag. If the flag is equal to a '0', the value of the displacement e is added to the Program Counter (PC) and the next instruction is fetched from the location designated by the new contents of the PC. The jump is measured from the address of the instruction opcode and has a range of −126 to + 129 bytes. The assembler calculates the displacement e and automatically adjusts for the twice incremented PC.

If the Zero Flag is equal to a '1', the next instruction to be executed is taken from the location following this instruction.

If the condition is met:
M CYCLES: 3 T STATES: 12(4,3,5) 4 MHz E.T.: 3.00

If the condition is not met:
M CYCLES: 2 T STATES: 764,3) 4 MHz E.T.: 1.75

Condition Bits Affected: None

Example:

The Zero Flag is reset and it is required to jump back 4 locations from 480. The assembly language statement is:

```
JR NZ,LABEL
```

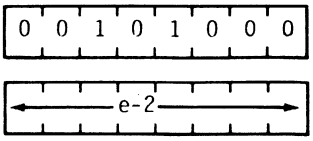
The resulting object code and final PC value is shown below:

Location				Opcode
47C	LABEL:	—		PC after jump
47D		—		
47E		—		
47F		—		
480		JR	NZ,LABEL	20
481				FA (2's complement-6)

Operation: If Z = 0, continue
If Z = 1, PC ← PC + e

Format:

Opcode	Operands
JR	Z,e



Description:

This instruction provides for conditional branching to other segments of a program depending on the results of a test on the Zero Flag. If the flag is equal to a '1', the value of the displacement e is added to the Program Counter (PC) and the next instruction is fetched from the location designated by the new contents of the PC. The jump is measured from the address of the instruction opcode and has a range of -126 to + 129 bytes. The assembler calculates the displacement e and automatically adjusts for the twice incremented PC.

If the Zero Flag is equal to a '0', the next instruction to be executed is taken from the location following this instruction.

If the condition is met:

M CYCLES: 3 T STATES: 1264,3,5) 4 MHz E.T.: 3.00

If the condition is not met:

M CYCLES: 2 T STATES: 7(4,3) 4 MHz E.T.: 1.75

Condition Bits Affected: None

Example:

The Zero Flag is set and it is required to jump forward 5 locations from address 300H. The following assembly language statement is used:

```
JR Z,LABEL
```

The resulting object code and final PC value is shown below:

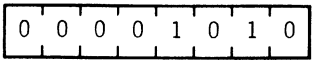
Location				Opcode
300		JR	Z,LABEL	28
301		—		03
302		—		
303		—		
304		—		
305	LABEL:	—		PC after jump

LD A,(BC)

Operation: A←(BC)

Format:

Opcode	Operands
LD	A,(BC)



0A

Description:

The contents of the memory location specified by the contents of the BC register pair are loaded into the Accumulator.

M CYCLES: 2 T STATES: 7(4,3) 4 MHz E.T.: 1.75

Condition Bits Affected: None

Example:

If the BC register pair contains the number 4747H, and memory address 4747H contains the byte 12H, then the instruction

```
LD A,(BC)
```

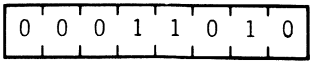
will result in byte 12H in register A.

LD A,(DE)

Operation: A←(DE)

Format:

Opcode	Operands
LD	A,(DE)



1A

Description:

The contents of the memory location specified by the register pair DE are loaded into the Accumulator.

M CYCLES: 2 T STATES: 7(4,3) 4 MHz E.T.: 1.75

Condition Bits Affected: None

Example:

If the DE register pair contains the number 30A2H and memory address 30A2H contains the byte 22H, then the instruction

```
LD A,(DE)
```

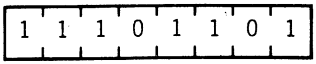
will result in byte 22H in register A.

LD A,I

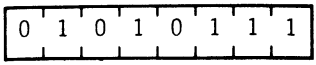
Operation: A←I

Format:

Opcode	Operands
LD	A,I



ED



57

Description:

The contents of the Interrupt Vector Register I are loaded into the Accumulator.

M CYCLES: 2 T STATES: 9(4,5) 4 MHz E.T.: 2.25

Condition Bits Affected:

S:	Set if I-Reg. is negative; reset otherwise
Z:	Set if I-Reg. is zero;reset otherwise
H:	Reset
P/V:	Contains contents of IFF2
N:	Reset
C:	Not affected

Example:

If the Interrupt Vector Register contains the byte 4AH, after the execution of

```
LD A,I
```

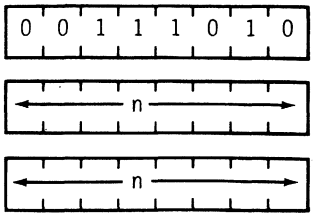
the accumulator will also contain 4AH.

LD A,(nn)

Operation: A←(nn)

Format:

Opcode	Operands
LD	A,(nn)



Description:

The contents of the memory location specified by the operands nn are loaded into the Accumulator. The first n operand is the low order byte of a two-byte memory address.

M CYCLES: 4 T STATES: 13(4,3,3,3) 4 MHz E.T.: 3.25

Condition Bits Affected: None

Example:

IF the contents of nn is number 8832H, and the content of memory address 8832H is byte 04H, after the instruction

LD A,(nn)

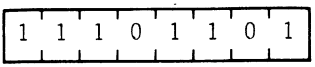
byte 04H will be in the Accumulator.

LD A,R

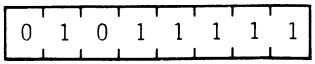
Operation: A←R

Format:

Opcode	Operands
LD	A,R



ED



5F

Description:

The contents of Memory Refresh Register R are loaded into the Accumulator.

M CYCLES: 2 T STATES: 9(4,5) 4 MHz E.T.: 2.25

Condition Bits Affected:

S: Set if R-Reg. is negative; reset otherwise
Z: Set if R-Reg. is zero; reset otherwise
H: Reset
P/V: Contains contents of IFF2
N: Reset
C: Not affected

Example:

If the Memory Refresh Register contains the byte 4AH, after the execution of

```
LD A,R
```

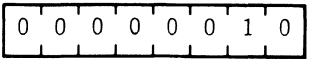
the Accumulator will also contain 4AH.

LD (BC),A

Operation: (BC)←A

Format:

Opcode	Operands
LD	(BC),A



02

Description:

The contents of the Accumulator are loaded into the memory location specified by the contents of the register pair BC.

M CYCLES: 2 T STATES: 7(4,3) 4 MHz E.T.: 1.75

Condition Bits Affected: None

Example:

If the Accumulator contains 7AH and the BC register pair contains 1212H the instruction

LD (BC),A

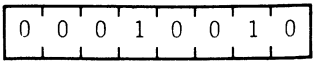
will result in 7AH being in memory location 1212H.

LD (DE),A

Operation: (DE)←A

Format:

Opcode	Operands
LD	(DE),A



12

Description:

The contents of the Accumulator are loaded into the memory location specified by the DE register pair.

M CYCLES: 2 T STATES: (7(4,3) 4 MHz E.T.: 1.75

Condition Bits Affected: None

Example:

If the contents of register pair DE are 1128H, and the Accumulator contains byte A0H, the instruction

```
LD (DE),A
```

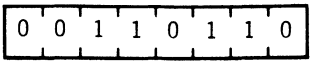
will result in A0H being in memory location 1128H.

LD (HL),n

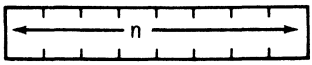
Operation: (HL)←n

Format:

Opcode	Operand
LD	(HL),n



36



Description:

Integer n is loaded into the memory address specified by the contents of the HL register pair.

M CYCLES: 3 T STATES: 10(4,3,3) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

If the HL register pair contains 4444H, the instruction

LD (HL),28H

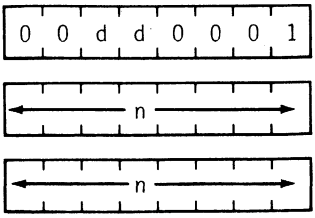
will result in the memory location 4444H containing the byte 28H.

LD dd,nn

Operation: dd←nn

Format:

Opcode	Operands
LD	dd,nn



Description:

The two-byte integer nn is loaded into the dd register pair, where dd defines the BC, DE, HL, or SP register pairs, assembled as follows in the object code:

Pair	dd
BC	00
DE	01
HL	10
SP	11

The first n operand in the assembled object code is the low order byte.

M CYCLES: 3 T STATES: 10(4,3,3) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

After the execution of

```
LD HL,5000H
```

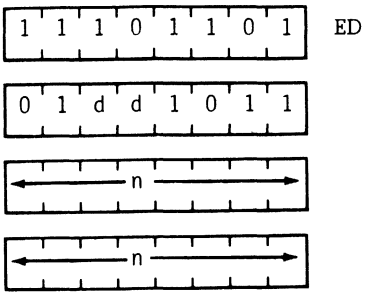
the contents of the HL register pair will be 5000H.

LD dd,(nn)

Operation: $dd_H \leftarrow (nn+1)$, $dd_L \leftarrow (nn)$

Format:

Opcode	Operands
LD	dd,(nn)



Description:

The contents of address nn are loaded into the low order portion of register pair dd, and the contents of the next highest memory address nn + 1 are loaded into the high order portion of dd. Register pair dd defines BC, DE, HL, or SP register pairs, assembled as follows in the object code:

Pair	dd
BC	00
DE	01
HL	10
SP	11

The first n operand in the assembled object code above is the low order byte of (nn).

M CYCLES: 6 T STATES: 20(4,4,3,3,3,3) 4 MHz E.T.: 5.00

Condition Bits Affected: None

Example:

If Address 2130H contains 65H and address 2131H contains 78H after the instruction

```
LD BC,(2130H)
```

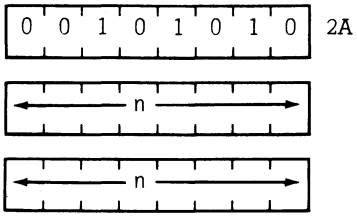
the BC register pair will contain 7865H.

LD HL,(nn)

Operation: H←(nn + 1), L←(nn)

Format:

Opcode	Operands
LD	HL,(nn)



Description:

The contents of memory address nn are loaded into the low order portion of register pair HL (register L), and the contents of the next highest memory address nn + 1 are loaded into the high order portion of HL (register H). The first n operand in the assembled object code above is the low order byte of nn.

M CYCLES: 5 T STATES: 16(4,3,3,3,3) 4 MHz E.T.: 4.00

Condition Bits Affected: None

Example:

If address 4545H contains 37H and address 4546H contains A1H after the instruction

```
LD HL,(4545H)
```

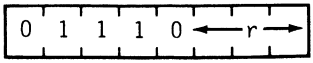
the HL register pair will contain A137H.

LD (HL),r

Operation: (HL)←r

Format:

Opcode	Operands
LD	(HL),r



Description:

The contents of register r are loaded into the memory location specified by the contents of the HL register pair. The symbol r identifies register A, B, C, D, E, H or L, assembled as follows in the object code:

Register	r
A	111
B	000
C	001
D	010
E	011
H	100
L	101

M CYCLES: 2 T STATES: 7(4,3) 4 MHz E.T.: 1.75

Condition Bits Affected: None

Example:

If the contents of register pair HL specifies memory location 2146H, and the B register contains the byte 29H, after the execution of

```
LD (HL),B
```

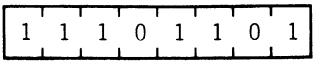
memory address 2146H will also contain 29H.

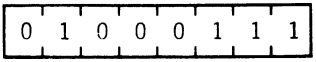
LD I,A

Operation: I←A

Format:

Opcode	Operands
LD	I,A





Description:

The contents of the Accumulator are loaded into the Interrupt Control Vector Register, I.

M CYCLES: 2 T STATES: 9(4,5) 4 MHz E.T.: 2.25

Condition Bits Affected: None

Example:

If the Accumulator contains the number 81H, after the instruction

LD I,A

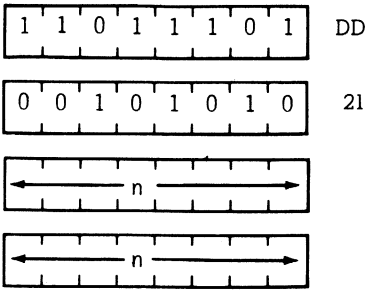
the Interrupt Vector Register will also contain 81H.

LD IX,nn

Operation: IX←nn

Format:

Opcode	Operands
LD	IX,nn



Description:

Integer nn is loaded into the Index Register IX. The first n operand in the assembled object code above is the low order byte.

M CYCLES: 4 T STATES: 14(4,4,3,3) 4 MHz E.T.: 3.50

Condition Bits Affected: None

Example:

After the instruction

```
LD IX,45A2H
```

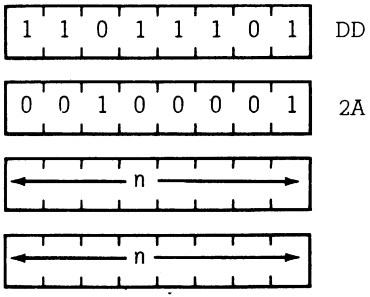
the Index Register will contain integer 45A2H.

LD IX,(nn)

Operation: $IX_H \leftarrow (nn+1)$, $IX_L \leftarrow (nn)$

Format:

Opcode	Operands
LD	IX,(nn)



Description:

The contents of the address nn are loaded into the low order portion of Index Register IX, and the contents of the next highest memory address nn + 1 are loaded into the high order portion of IX. The first n operand in the assembled object code above is the low order byte of nn.

M CYCLES: 6 T STATES: 20(4,4,3,3,3,3) 4 MHz E.T.: 5.00

Condition Bits Affected: None

Example:

If address 6666H contains 92H and address 6667H contains DAH, after the instruction

LD IX,(6666H)

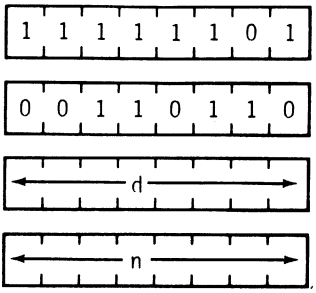
the Index Register IX will contain DA92H.

LD (IX+d),n

Operation: (IX + d)←n

Format:

Opcode	Operands
LD	(IX + d),n



Description:

The n operand is loaded into the memory address specified by the sum of the contents of the Index Register IX and the two's complement displacement operand d.

M CYCLES: 5 T STATES: 19(4,4,3,5,3) 4 MHz E.T.: 4.75

Condition Bits Affected: None

Example:

If the Index Register IX contains the number 219AH the instruction

LD (IX + 5H),5AH

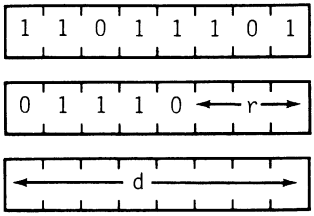
would result in the byte 5AH in the memory address 219FH.

LD (IX+d),r

Operation: (IX + d)←r

Format:

Opcode	Operands
LD	(IX + d),r



Description:

The contents of register r are loaded into the memory address specified by the contents of Index Register IX summed with d, a two's complement displacement integer. The symbol r identifies register A, B, C, D, E, H or L, assembled as follows in the object code:

Register	r
A	111
B	000
C	001
D	010
E	011
H	100
L	101

M CYCLES: 5 T STATES: 19(4,4,3,5,3) 4 MHz E.T.: 4.75

Condition Bits Affected: None

Example:

If the C register contains the byte 1CH, and the Index Register IX contains 3100H, then the instruction

LD (IX + 6H),C

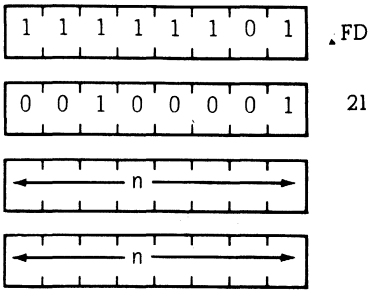
will perform the sum 3100H + 6H and will load 1CH into memory location 3106H.

LD IY,nn

Operation: IY←nn

Format:

Opcode	Operands
LD	IY,nn



Description:

Integer nn is loaded into the Index Register IY. The first n operand in the assembled object code above is the low order byte.

M CYCLES: 4 T STATES: 14(4,4,3,3) 4 MHz E.T.: 3.50

Condition Bits Affected: None

Example:

After the instruction:

```
LD IY,7733H
```

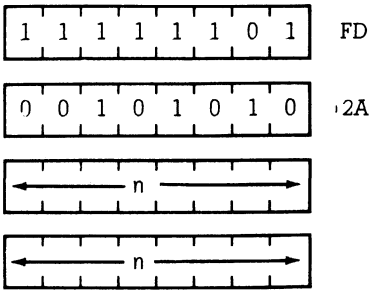
the Index Register IY will contain the integer 7733H.

LD IY,(nn)

Operation: $IY_H \leftarrow (nn+1)$, $IY_L \leftarrow (nn)$

Format:

Opcode	Operands
LD	IY,(nn)



Description:

The contents of address nn are loaded into the low order portion of Index Register IY, and the content the next highest memory address nn + 1 are loaded in the high order portion of IY. The first n operand in assembled object code above is the low order byte of nn.

M CYCLES: 6 T STATES: 20(4,4,3,3,3,3) 4 MHz E.T.: 5

Condition Bits Affected: None

Example:

If address 6666H contains 92H and address 6667H contains DAH, after the instruction

```
LD IY,(6666H)
```

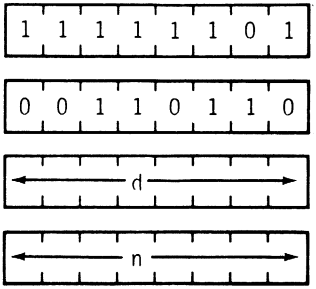
the Index Register IY will contain DA92H.

LD (IY+d),n

Operation: (IY + d)←n

Format:

Opcode	Operands
LD	(IY + d),n



Description:

Integer n is loaded into the memory location specified by the contents of the Index Register summed with a displacement integer d.

M CYCLES: 5 T STATES: 19(4,4,3,5,3) 4 MHz E.T.: 4.75

Condition Bits Affected: None

Example:

If the Index Register IY contains the number A940H, the instruction

LD (IY + 10H),97H

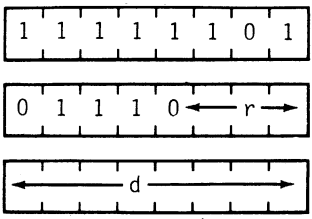
would result in byte 97H in memory location A950H.

LD (IY+d),r

Operation: (IY + d)←r

Format:

Opcode	Operands
LD	(IY + d),r



Description:

The contents of register r are loaded into the memory address specified by the sum of the contents of the Index Register IY and d, a two's complement displacement integer. The symbol r is specified according to the following table.

Register	r
A	111
B	000
C	001
D	010
E	011
H	100
L	101

M CYCLES: 5 T STATES: 19(4,4,3,5,3) 4 MHz E.T.: 4.75

Condition Bits Affected: None

Example:

If the C register contains the byte 48H, and the Index Register IY contains 2A11H, then the instruction

LD (IY + 4H),C

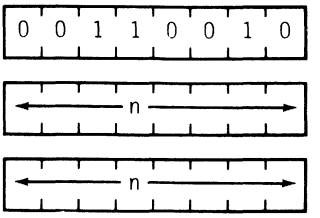
will perform the sum 2A11H + 4H, and will load 48H into memory location 2A15H.

LD (nn),A

Operation: (nn)←A

Format:

Opcode	Operands
LD	(nn),A



Description:

The contents of the Accumulator are loaded into the memory address specified by the operands nn. The first n operand in the assembled object code above is the low order byte of nn.

M CYCLES: 4 T STATES: 13(4,3,3,3) 4 MHz E.T.: 3.25

Condition Bits Affected: None

Example:

If the contents of the Accumulator are byte D7H, after the execution of

LD (3141H),A

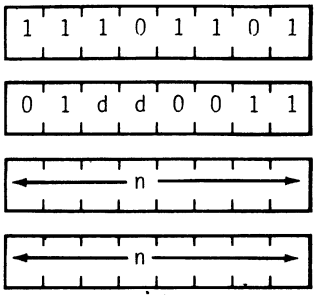
D7H will be in memory location 3141H.

LD (nn),dd

Operation: $(nn+1) \leftarrow dd_H$, $(nn) \leftarrow dd_L$

Format:

Opcode	Operands
LD	(nn),dd



Description:

The low order byte of register pair dd is loaded into memory address nn; the upper byte is loaded into memory address nn + 1. Register pair dd defines either BC, DE, HL, or SP, assembled as follows in the object code:

Pair	dd
BC	00
DE	01
HL	10
SP	11

The first n operand in the assembled object code is the low order byte of a two byte memory address.

M CYCLES: 6 T STATES: 20(4,4,3,3,3,3) 4 MHz E.T.: 5.00

Condition Bits Affected: None

Example:

If register pair BC contains the number 4644H, the instruction

LD (1000H),BC

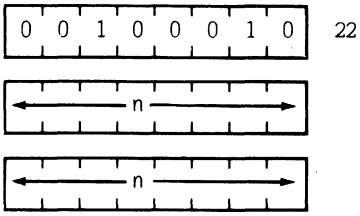
will result in 44H in memory location 1000H, and 46H in memory location 1001H.

LD (nn),HL

Operation: (nn + 1)←H (nn)←L

Format:

Opcode	Operands
LD	(nn),HL



Description:

The contents of the low order portion of register pair HL (register L) are loaded into memory address nn, and the contents of the high order portion of HL (register H) are loaded into the next highest memory address nn + 1. The first n operand in the assembled object code above is the low order byte of nn.

M CYCLES: 5 T STATES: 1664,3,3,3,3) 4 MHz E.T.: 4.00

Condition Bits Affected: None

Example:

If the content of register pair HL is 483AH, after the instruction

LD (B229H),HL

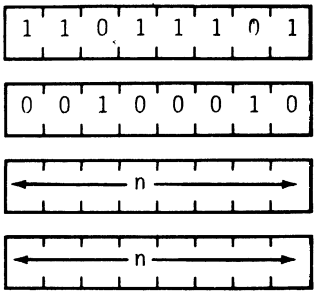
address B229H will contain 3AH, and address B22AH will contain 48H.

LD (nn),IX

Operation: $(nn+1) \leftarrow IX_H$, $(nn) \leftarrow IX_L$

Format:

Opcode	Operands
LD	(nn),IX



Description:

The low order byte in Index Register IX is loaded into memory address nn; the upper order byte is loaded into the next highest address nn + 1. The first n operand in the assembled object code above is the low order byte of nn.

M CYCLES: 6 T STATES: 20(4,4,3,3,3,3) 4 MHz E.T.: 5.00

Condition Bits Affected: None

Example:

If the Index Register IX contains 5A30H, after the instruction

LD (4392H),IX

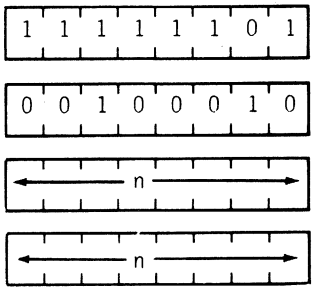
memory location 4392H will contain number 30H and location 4393H will contain 5AH.

LD (nn),IY

Operation: $(nn+1) \leftarrow IY_H$, $(nn) \leftarrow IY_L$

Format:

Opcode	Operands
LD	(nn),IY



Description:

The low order byte in Index Register IY is loaded into memory address nn; the upper order byte is loaded into memory location nn + 1. The first n operand in the assembled object code above is the low order byte of nn.

M CYCLES: 6 T STATES: 20(4,4,3,3,3,3) 4 MHz E.T.: 5.00

Condition Bits Affected: None

Example:

If the Index Register IY contains 4174H after the instruction

```
LD (8838H),IY
```

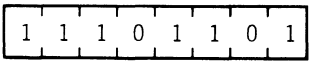
memory location 8838H will contain number 74H and memory location 8839H will contain 41H.

LD R,A

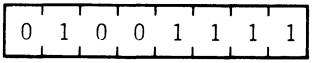
Operation: R ← A

Format:

Opcode	Operands
LD	R,A



ED



4F

Description:

The contents of the Accumulator are loaded into the Memory Refresh register R.

M CYCLES: 2 T STATES: 9(4,5) 4 MHz E.T.: 2.25

Condition Bits Affected: None

Example:

If the Accumulator contains the number B4H, after the instruction

LD R,A

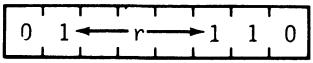
the Memory Refresh Register will also contain B4H.

LD r,(HL)

Operation: r←(HL)

Format:

Opcode	Operands
LD	r,(HL)



Description:

The eight-bit contents of memory location (HL) are loaded into register r, where r identifies register A, B, C, D, E, H or L, assembled as follows in the object code:

Register	r
A	111
B	000
C	001
D	010
E	011
H	100
L	101

M CYCLES: 2 T STATES: 7(4,3) 4 MHz E.T.: 1.75

Condition Bits Affected: None

Example:

If register pair HL contains the number 75A1H, and memory address 75A1H contains the byte 58H, the execution of

```
LD C,(HL)
```

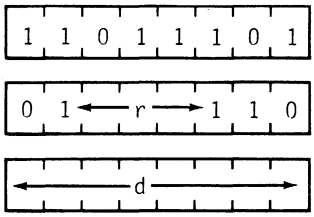
will result in 58H in register C.

LD r,(IX+d)

Operation: r←(IX + d)

Format:

Opcode	Operands
LD	r,(IX + d)



Description:

The operand (IX + d) (the contents of the Index Register IX summed with a displacement integer d) is loaded into register r, where r identifies register A, B, C, D, E, H or L, assembled as follows in the object code:

Register	r
A	111
B	000
C	001
D	010
E	011
H	100
L	101

M CYCLES: 5 T STATES: 19(4,4,3,5,3) 4 MHz E.T.: 4.75

Condition Bits Affected: None

Example:

If the Index Register IX contains the number 25AFH, the instruction

LD B,(IX + 19H)

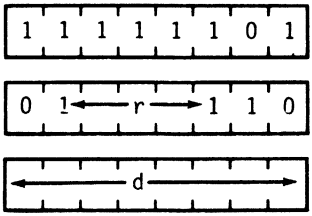
will cause the calculation of the sum 25AFH + 19H, which points to memory location 25C8H. If this address contains byte 39H, the instruction will result in register B also containing 39H.

LD r,(IY+d)

Operation: r←(IY + d)

Format:

Opcode	Operands
LD	r,(IY + d)



Description:

The operand (IY + d) (the contents of the Index Register IY summed with a displacement integer d) is loaded into register r, where r identifies register A, B, C, D, E, H or L, assembled as follows in the object code:

Register	r
A	111
B	000
C	001
D	010
E	011
H	100
L	101

M CYCLES: 5 T STATES: 19(4,4,3,5,3) 4 MHz E.T.: 4.75

Condition Bits Affected: None

Example:

If the Index Register IY contains the number 25AFH, the instruction

LD B,(IY + 19H)

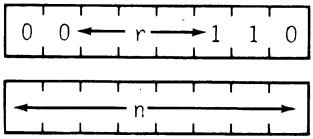
will cause the calculation of the sum 25AFH + 19H, which points to memory location 25C8H. If this address contains byte 39H, the instruction will result in register B also containing 39H.

LD r,n

Operation: r←n

Format:

Opcode	Operands
LD	r,n



Description:

The eight-bit integer n is loaded into any register r, where r identifies register A, B, C, D, E, H or L, assembled as follows in the object code:

Register	r
A	111
B	000
C	001
D	010
E	011
H	100
L	101

M CYCLES: 2 T STATES: 7(4,3) 4 MHz E.T.: 1.75

Condition Bits Affected: None

Example:

After the execution of

```
LD E,A5H
```

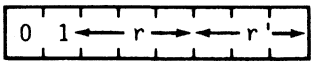
the contents of register E will be A5H.

LD r,r'

Operation: r←r'

Format:

Opcode	Operands
LD	r,r'



Description:

The contents of any register r' are loaded into any other register r. Note: r,r' identifies any of the registers A, B, C, D, E, H, or L, assembled as follows in the object code:

Register	r,r'
A	111
B	000
C	001
D	010
E	011
H	100
L	101

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.0

Condition Bits Affected: None

Example:

If the H register contains the number 8AH, and the E register contains 10H, the instruction

```
LD H,E
```

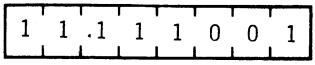
would result in both registers containing 10H.

LD SP,HL

Operation: SP←HL

Format:

Opcode	Operands
LD	SP,HL



F9

Description:

The contents of the register pair HL are loaded into the Stack Pointer SP.

M CYCLES: 1 T STATES: 6 4 MHz E.T.: 1.50

Condition Bits Affected: None

Example:

If the register pair HL contains 442EH, after the instruction

```
LD SP,HL
```

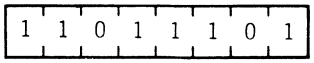
the Stack Pointer will also contain 442EH.

LD SP,IX

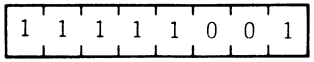
Operation: SP←IX

Format:

Opcode	Operands
LD	SP,IX



DD



F9

Description:

The two byte contents of Index Register IX are loaded into the Stack Pointer SP.

M CYCLES: 2 T STATES: 10(4,6) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

If the contents of the Index Register IX are 98DAH, after the instruction

LD SP,IX

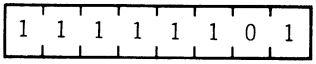
the contents of the Stack Pointer will also be 98DAH.

LD SP,IY

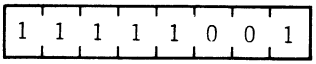
Operation: SP←IY

Format:

Opcode	Operands
LD	SP,IY



FD



F9

Description:

The two byte contents of Index Register IY are loaded into the Stack Pointer SP.

M CYCLES: 2 T STATES: 10(4,6) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

If Index Register IY contains the integer A227H, after the instruction

LD SP,IY

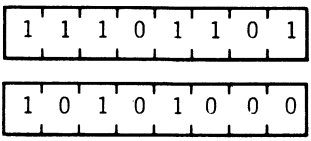
the Stack Pointer will also contain A227H.

Operation: (DE)←(HL), DE←DE-1, HL←HL-1, BC←BC-1

Format:

Opcode

LDD



Description:

This two byte instruction transfers a byte of data from the memory location addressed by the contents of the HL register pair to the memory location addressed by the contents of the DE register pair. Then both of these register pairs including the BC (Byte Counter) register pair are decremented.

M CYCLES: 4 T STATES: (16(4,4,3,5) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Set if BC-1≠0; reset otherwise
N: Reset
C: Not affected

Example:

If the HL register pair contains 1111H, memory location 1111H contains the byte 88H, the DE register pair contains 2222H, memory location 2222H contains byte 66H, and the BC register pair contains 7H, then the instruction

LDD

will result in the following contents in register pairs and memory addresses:

HL : 1110H
(1111H) : 88H
DE : 2221H
(2222H) : 88H
BC : 6H

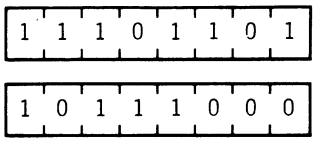
LDDR

Operation: (DE)←(HL), DE←DE-1, HL←HL-1, BC←BC-1

Format:

Opcode

LDDR



Description:

This two byte instruction transfers a byte of data from the memory location addressed by the contents of the DE register pair. Then both of these registers as well as the BC (Byte Counter) are decremented. If decrementing causes the BC to go to zero, the instruction is terminated. If BC is not zero, the program counter is decremented by 2 and the instruction is repeated. Note that if BC is set to zero prior to instruction execution, the instruction will loop through 64K bytes. Interrupts will be recognized and two refresh cycles will be executed after each data transfer.

For BC≠0:

M CYCLES: 5 T STATES: 21(4,4,3,5,5) 4 MHz E.T.: 5.25

For BC = 0:

M CYCLES: 4 T STATES: 16(4,4,3,5) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Reset
N: Reset
C: Not affected

Example:

If the HL register pair contains 1114H, the DE register pair contains 2225H, the BC register pair contains 0003H, and memory locations have these contents:

(1114H) : A5H (2225H) : C5H
(1113H) : 36H (2224H) : 59H
(1112H) : 88H (2223H) : 66H

then after the execution of

LDDR

the contents of register pairs and memory locations will be:

HL : 1111H
DE : 2222H
BC : 0000H

(1114H) : A5H (2225H) : A5H
(1113H) : 36H (2224H) : 36H
(1112H) : 88H (2223H) : 88H

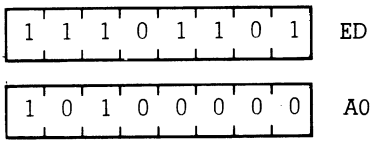
LDI

Operation: (DE)←(HL), DE←DE+1, HL←HL+1, BC←BC-1

Format:

Opcode

LDI



Description:

A byte of data is transferred from the memory location addressed by the contents of the HL register pair to the memory location addressed by the contents of the DE register pair. Then both these register pairs are incremented and the BC (Byte Counter) register pair is decremented.

M CYCLES: 4 T STATES: 16(4,4,3,5) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Set if BC-1≠0; reset otherwise
N: Reset
C: Not affected

Example:

If the HL register pair contains 1111H, memory location 1111H contains the byte 88H, the DE register pair contains 2222H, the memory location 2222H contains byte 66H, and the BC register pair contains 7H, then the instruction

```
LDI
```

will result in the following contents in register pairs and memory addresses:

```
  HL  : 1112H
(1111H) :  88H
  DE  : 2223H
(2222H) :  88H
  BC  :   6H
```

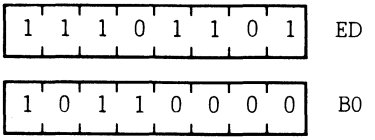
LDIR

Operation: (DE)←(HL), DE←DE+1, HL←HL+1, BC←BC-1

Format:

Opcode

LDIR



Description:

This two byte instruction transfers a byte of data from the memory location addressed by the contents of the HL register pair to the memory location addressed by the DE register pair. Then both these register pairs are incremented and the BC (Byte Counter) register pair is decremented. If decrementing causes the BC to go to zero, the instruction is terminated. If BC is not zero the program counter is decremented by 2 and the instruction is repeated. Note that if BC is set to zero prior to instruction execution, the instruction will loop through 64K bytes. Interrupts will be recognized and two refresh cycles will be executed after each data transfer.

For BC≠0:

M CYCLES: 5 T STATES: 21(4,4,3,5,5) 4 MHz E.T.: 5.25

For BC=0:

M CYCLES: 4 T STATES: 16(4,4,3,5) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Reset
N: Reset
C: Not affected

Example:

If the HL register pair contains 1111H, the DE register pair contains 2222H, the BC register pair contains 0003H, and memory locations have these contents:

(1111H) : 88H (2222H) : 66H
(1112H) : 36H (2223H) : 59H
(1113H) : A5H (2224H) : C5H

then after the execution of

```
LDIR
```

the contents of register pairs and memory locations will be:

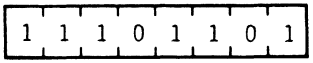
```
HL : 1114H
DE : 2225H
BC : 0000H

(1111H) : 88H        (2222H) : 88H
(1112H) : 36H        (2223H) : 36H
(1113H) : A5H        (2224H) : A5H
```

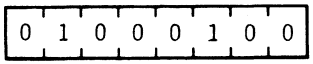
Operation: A←0-A

Format:

Opcode

NEG





Description:

Contents of the Accumulator are negated (two's complement). This is the same as subtracting the contents of the Accumulator from zero. Note that 80H is left unchanged.

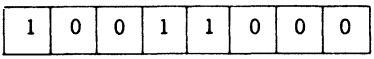
M CYCLES: 2 T STATES: 8(4,4) 4 MHz E.T.: 2.00

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Set if there is a borrow and reset otherwise.
P/V: Set if Acc. was 80H before operation; reset otherwise
N: Set
C: Set if Acc. was not 00H before operation; reset otherwise

Example:

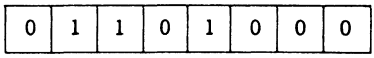
If the contents of the Accumulator are



after the execution of

NEG

the Accumulator contents will be



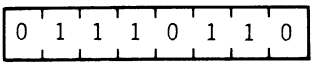
NOP

Operation: no operation

Format:

Opcode

NOP



00

Description:

CPU performs no operation during this machine cycle.

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected: None

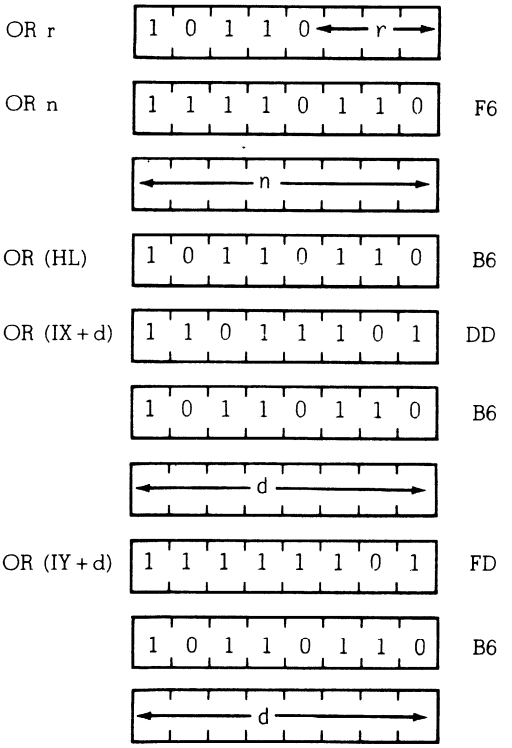
OR s

Operation: A←A ∨ s

Format:

Opcode	**Operands**
OR	s

The s operand is any of r,n,(HL), (IX + d) or (IY + d), as defined for the analogous ADD instructions. These various possible opcode-operand combinations are assembled as follows in the object code:



r identifies registers B,C,D,E,H,L or A assembled as follows in the object code field above:

OR s

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

A logical OR operation, bit by bit, is performed between the byte specified by the s operand and the byte contained in the Accumulator; the result is stored in the Accumulator.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
OR r	1	4	1.00
OR n	2	7(4,3)	1.75
OR (HL)	2	7(4,3)	1.75
OR (IX+d)	5	19(4,4,3,5,3)	4.75
OR (IY+d)	5	19(4,4,3,5,3)	4.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set
P/V:	Set if parity even; reset otherwise
N:	Reset
C:	Reset

Example:

If the H register contains 48H (01001000) and the Accumulator contains 12H (00010010) after the execution of

```
OR H
```

the Accumulator will contain 5AH (01011010).

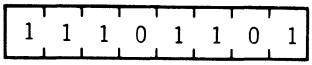
OTDR

Operation: (C)←(HL), B←B-1, HL←HL-1

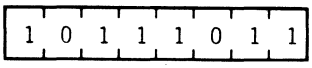
Format:

Opcode

OTDR



ED



BB

Description:

The contents of the HL register pair are placed on the address bus to select a location in memory. The byte contained in this memory location is temporarily stored in the CPU. Then, after the byte counter (B) is decremented, the contents of register C are placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. Register B may be used as a byte counter, and its decremented value is placed on the top half (A8 through A15) of the address bus at this time. Next the byte to be output is placed on the data bus and written into the selected peripheral device. Then register pair HL is decremented and if the decremented B register is not zero, the Program Counter (PC) is decremented by 2 and the instruction is repeated. If B has gone to zero, the instruction is terminated. Note that if B is set to zero prior to instruction execution, the instruction will output 256 byte of data. Interrupts will be recognized and two refresh cycles will be executed after each data transfer.

If B≠0:

M CYCLES: 5 T STATES: (21(4,5,3,4,5) 4 MHz E.T.: 5.25

If B=0:

M CYCLES: 4 T STATES: 16(4,5,3,4) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Unknown
Z: Set
H: Unknown
P/V: Unknown
N: Set
C: Not affected

OTDR

Example:

If the contents of register C are 07H, the contents of register B are 03H, the contents of the HL register pair are 1000H, and memory locations have the following contents:

Location	Contents
0FFEH	51H
0FFFH	A9H
1000H	03H

then after the execution of

```
OTDR
```

the HL register pair will contain OFFDH, register B will contain zero, and a group of bytes will have been written to the peripheral device mapped to I/O port address 07H in the following sequence:

```
03H
A9H
51H
```

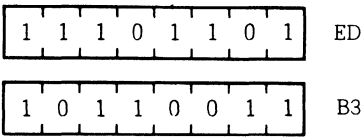
OTIR

Operation: (C)←(HL), B←B-1, HL←HL+1

Format:

Opcode

OTIR



Description:

The contents of the HL register pair are placed on the address bus to select a location in memory. The byte contained in this memory location is temporarily stored in the CPU. Then, after the byte counter (B) is decremented, the contents of register C are placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. Register B may be used as a byte counter, and its decremented value is placed on the top half (A8 through A15) of the address bus at this time. Next the byte to be output is placed on the data bus and written into the selected peripheral device. Then register pair HL is incremented. If the decremented B register is not zero, the Program Counter (PC) is decremented by 2 and the instruction is repeated. If B has gone to zero, the instruction is terminated. Note that if B is set to zero prior to instruction execution, the instruction will output 256 bytes of data. Interrupts will be recognized and two refresh cycles will be executed after each data transfer.

If B≠0:

M CYCLES: 5 T STATES: 21(4,5,3,4,5) 4 MHz E.T.: 5.25

If B=0:

M CYCLES: 4 T STATES: 16(4,5,3,4) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Unknown
Z: Set
H: Unknown
P/V: Unknown
N: Set
C: Not affected

OTIR

Example:

If the contents of register C are 07H, the contents of register B are 03H, the contents of the HL register pair are 1000H, and memory locations have the following contents:

Locations	**Contents**
1000H	51 H
1001H	A9 H
1002H	03 H

then after the execution of

```
OTIR
```

the HL register pair will contain 1003H, register B will contain zero, and a group of bytes will have been written to the peripheral device mapped to I/O port address 07H in the following sequence:

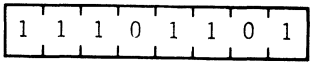
```
51 H
A9 H
03 H
```

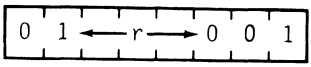
OUT (C),r

Operation: (C)←r

Format:

Opcode	Operands
OUT	(C),r





Description:

The contents of register C are placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. The contents of Register B are placed on the top half (A8 through A15) of the address bus at this time. Then the byte contained in register r is placed on the data bus and identifies any of the CPU registers shown in the following table, which also shows the corresponding 3-bit "r" field for each which appears in the assembled object code:

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

M CYCLES: 3 T STATES: 12(4,4,4) 4 MHz E.T.: 3.00

Condition Bits Affected: None

Example:

If the contents of register C are 01H and the contents of register D are 5AH, after the execution of

```
OUT (C),D
```

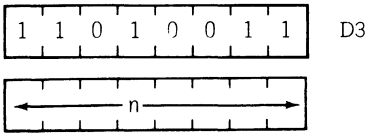
the byte 5AH will have been written to the peripheral device mapped to I/O port address 01H.

OUT (n),A

Operation: (n)←A

Format:

Opcode	Operands
OUT	(n),A



Description:

The operand n is placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. The contents of the Accumulator (register A) also appear on the top half (A8 through A15) of the address bus at this time. Then the byte contained in the Accumulator is placed on the data bus and written into the selected peripheral device.

M CYCLES: 3 T STATES: 11(4,3,4) 4 MHz E.T.: 2.75

Condition Bits Affected: None

Example:

If the contents of the Accumulator are 23H, then after the execution of

```
OUT (01H),A
```

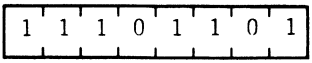
the byte 23H will have been written to the peripheral device mapped to I/O port address 01H.

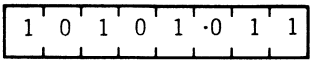
Operation: (C)←(HL), B←B-1, HL←HL-1

Format:

Opcode

OUTD





Description:

The contents of the HL register pair are placed on the address bus to select a location in memory. The byte contained in this memory location is temporarily stored in the CPU. Then, after the byte counte (B) is decremented, the contents of register C are placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. Register B may be used as a byte counter, and its decremented value is placed on the top half (A8 through A15) of the address bus at this time. Next the byte to be output is placed on the data bus and written into the selected peripheral device. Finally the register pair HL is decremented.

M CYCLES: 4 T STATES: 16(4,5,3,4) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Unknown
Z: Set if B-1 = 0; reset otherwise
H: Unknown
P/V: Unknown
N: Set
C: Not affected

Example:

If the contents of register C are 07H, the contents of register B are 10H, the contents of the HL register pair are 1000H, and the contents of memory location 1000H are 59H, after the execution of

```
OUTD
```

register B will contain 0FH, the HL register pair will contain 0FFFH, and the byte 59H will have been written to the peripheral device mapped to I/0 port address 07H.

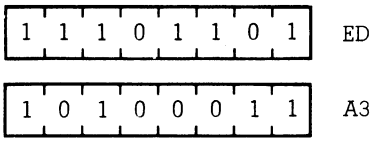
OUTI

Operation: (C)←(HL), B←B-1, HL←HL+1

Format:

Opcode

OUTI



Description:

The contents of the HL register pair are placed on the address bus to select a location in memory. The byte contained in this memory location is temporarily stored in the CPU. Then, after the byte counter (B) is decremented, the contents of register C are placed on the bottom half (A0 through A7) of the address bus to select the I/0 device at one of 256 possible ports. Register B may be used as a byte counter, and its decremented value is placed on the top half (A8 through A15) of the address bus. The byte to be output is placed on the data bus and written into selected peripheral device. Finally the register pair HL is incremented.

M CYCLES: 4 T STATES: 16(4,5,3,4) 4 MHz E.T.: 4.00

Condition Bits Affected:

S: Unknown
Z: Set if B-1 = 0; reset otherwise
H: Unknown
P/V: Unknown
N: Set
C: Not affected

Example:

If the contents of register C are 07H, the contents of register B are 10H, the contents of the HL register pair are 1000H, and the contents of memory address 1000H are 59H, then after the execution of

```
OUTI
```

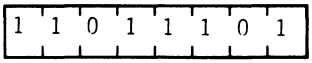
register B will contain 0FH, the HL register pair will contain 1001H, and the byte 59H will have been written to the peripheral mapped to I/O port address 07H.

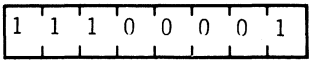
POP IX

Operation: $IX_H \leftarrow (SP + 1)$, $IX_L \leftarrow (SP)$

Format:

Opcode	Operands
POP	IX





Description:

The top two bytes of the external memory LIFO (last-in, first-out) Stack are popped into Index Register IX. The Stack Pointer (SP) register pair holds the 16-bit address of the current "top" of the Stack. This instruction first loads into the low order portion of IX the byte at the memory location corresponding to the contents of SP; then SP is incremented and the contents of the corresponding adjacent memory location are loaded into the high order portion of IX. The SP is now incremented again.

M CYCLES: 4 T STATES: 14(4,4,3,3) 4 MHz E.T.: 3.50

Condition Bits Affected: None

Example:

If the Stack Pointer contains 1000H, memory location 1000H contains 55H, and location 1001H contains 33H, the instruction

```
POP IX
```

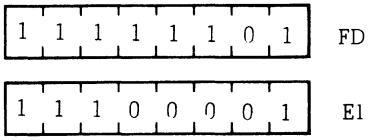
will result in Index Register IX containing 3355H, and the Stack Pointer containing 1002H.

POP IY

Operation: $IY_H \leftarrow (SP+1)$, $IY_L \leftarrow (SP)$

Format:

Opcode	Operands
POP	IY



Description:

The top two bytes of the external memory LIFO (last-in, first-out) Stack are popped into Index Register IY. The Stack Pointer (SP) register pair holds the 16-bit address of the current "top" of the Stack. This instruction first loads into the low order portion of IY the byte at the memory location corresponding to the contents of SP; then SP is incremented and the contents of the corresponding adjacent memory location are loaded into the high order portion of IY. The SP is now incremented again.

M CYCLES: 4 T STATES: 14(4,4,3,3) 4 MHz E.T.: 3.50

Condition Bits Affected: None

Example:

If the Stack Pointer contains 1000H, memory location 1000H contains 55H, and location 1001H contains 33H, the instruction

```
POP IY
```

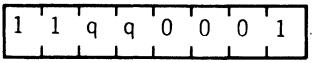
will result in Index Register IY containing 3355H, and the Stack Pointer containing 1002H.

POP qq

Operation: $qq_H \leftarrow (SP+1)$, $qq_L \leftarrow (SP)$

Format:

Opcode	Operands
POP	qq



Description:

The top two bytes of the external memory LIFO (last-in, first-out) Stack are popped into register pair qq. The Stack Pointer (SP) register pair holds the 16-bit address of the current "top" of the Stack. This instruction first loads into the low order portion of qq, the byte at the memory location corresponding to the contents of SP; then SP is incremented and the contents of the corresponding adjacent memory location are loaded into the high order portion of qq and the SP is now incremented again. The operand qq defines register pair BC, DE, HL, or AF, assembled as follows in the object code:

Pair	qq
BC	00
DE	01
HL	10
AF	11

M CYCLES: 3 T STATES: 10(4,3,3) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

If the Stack Pointer contains 1000H, memory location 1000H contains 55H, and location 1001H contains 33H, the instruction

```
POP HL
```

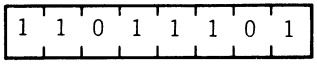
will result in register pair HL containing 3355H, and the Stack Pointer containing 1002H.

PUSH IX

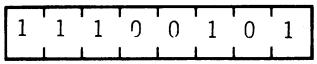
Operation: (SP-2)←IX_L, (SP-1)←IX_H

Format:

Opcode	Operands
PUSH	IX



DD



E5

Description:

The contents of the Index Register IX are pushed into the external memory LIFO (last-in, first-out) Stack. The Stack Pointer (SP) register pair holds the 16-bit address of the current "top" of the Stack. This instruction first decrements the SP and loads the high order byte of IX into the memory address now specified by the SP; then decrements the SP again and loads the low order byte into the memory location corresponding to this new address in the SP.

M CYCLES: 3 T STATES: 15(4.5.3.3) 4 MHz E.T.: 3.75

Condition Bits Affected: None

Example:

If the Index Register IX contains 2233H and the Stack Pointer contains 1007H, after the instruction

```
PUSH IX
```

memory address 1006H will contain 22H, memory address 1005H will contain 33H, and the Stack Pointer will contain 1005H.

PUSH IY

Operation: $(SP-2) \leftarrow IY_L$, $(SP-1) \leftarrow IY_H$

Format:

Opcode	Operands	
PUSH	IY	
1 1 1 1 1 1 0 1		FD
1 1 1 0 0 1 0 1		E5

Description:

The contents of the Index Register IY are pushed into the external memory LIFO (last-in, first-out) Stack. The Stack Pointer (SP) register pair holds the 16-bit address of the current "top" of the Stack. This instruction first decrements the SP and loads the high order byte of IY into the memory address now specified by the SP; then decrements the SP again and loads the low order byte into the memory location corresponding to this new address in the SP.

M CYCLES: 4 T STATES: 15(4,5,3,3) 4 MHz E.T.: 3.75

Condition Bits Affected: None

Example:

If the Index Register IY contains 2233H and the Stack Pointer contains 1007H, after the instruction

```
PUSH IY
```

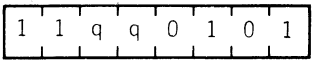
memory address 1006H will contain 22H, memory address 1005H will contain 33H, and the Stack Pointer will contain 1005H.

PUSH qq

Operation: $(SP-2) \leftarrow qq_L$, $(SP-1) \leftarrow qq_H$

Format:

Opcode	Operands
PUSH	qq



Description:

The contents of the register pair qq are pushed into the external memory LIFO (last-in, first-out) Stack. The Stack Pointer (SP) register pair holds the 16-bit address of the current "top" of the Stack. This instruction first decrements the SP and loads the high order byte of register pair qq into the memory address now specified by the SP; then decrements the SP again and loads the low order byte of qq into the memory location corresponding to this new address in the SP. The operand qq means register pair BC, DE, HL, or AF, assembled as follows in the object code:

Pair	qq
BC	00
DE	01
HL	10
AF	11

M CYCLES: 3 T STATES: 11(5,3,3) 4 MHz E.T.: 2.75

Condition Bits Affected: None

Example:

If the AF register pair contains 2233H and the Stack Pointer contains 1007H, after the instruction

```
PUSH AF
```

memory address 1006H will contain 22H, memory address 1005H will contain 33H, and the Stack Pointer will contain 1005H.

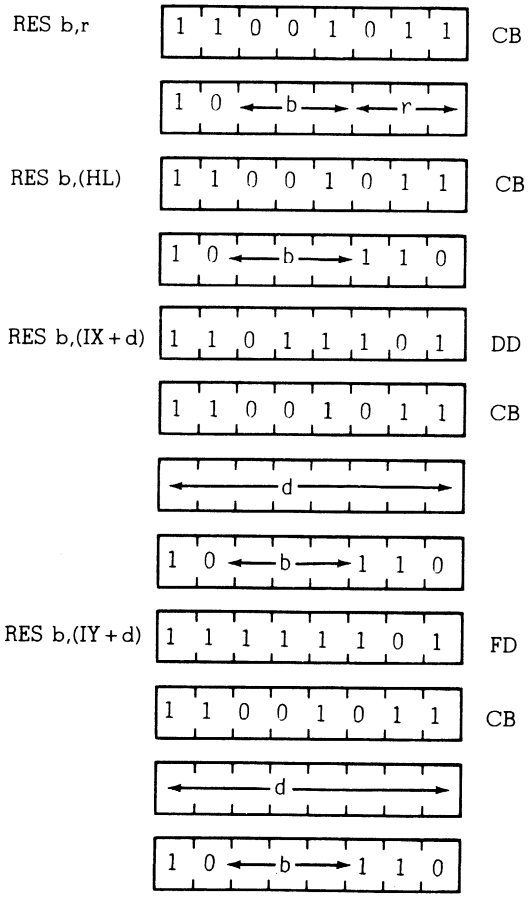
RES b,m

Operation: $s_b \leftarrow 0$

Format:

Opcode	Operands
RES	b,m

Operand b is any bit (7 through 0) of the contents of the m operand, (any of r, (HL), (IX + d) or (IY + d) as defined for the analogous SET instructions. These various possible opcode-operand combinations are assembled as follows in the object code:



RES b,m

Bit Reset	b	Register	r
0	000	B	000
1	001	C	001
2	010	D	010
3	011	E	011
4	100	H	100
5	101	L	101
6	110	L	111
7	111		

Description:

Bit b in operand m is reset.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
RES r	4	8(4,4)	2.00
RES (HL)	4	15(4,4,4,3)	3.75
RES (IX + d)	6	23(4,4,3,5,4,3)	5.75
RES (IY + d)	6	23(4,4,3,5,4,3)	5.75

Condition Bits Affected: None

Example:

After the execution of

```
RES 6,D
```

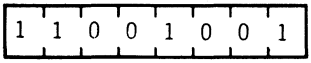
bit 6 in register D will be reset. (Bit 0 in register D is the least significant bit.)

Operation: $PC_L \leftarrow (SP)$, $PC_H \leftarrow (SP + 1)$

Format:

Opcode

RET



C9

Description:

Control is returned to the original program flow by popping the previous contents of the Program Counter (PC) off the top of the external memory stack, where they were pushed by the CALL instruction. This is accomplished by first loading the low-order byte of the PC with the contents of the memory address pointed to by the Stack Pointer (SP), then incrementing the SP and loading the high-order byte of the PC with the contents of the memory address now pointed to by the SP. (The SP is now incremented a second time). On the following machine cycle the CPU will fetch the next program opcode from the location in memory now pointed to by the PC.

M CYCLES: 3 T STATES: 10(4,3,3) 4 MHz E.T.: 2.50

Condition Bits Affected: None

Example:

If the contents of the Program Counter are 3535H, the contents of the Stack Pointer are 2000H, the contents of memory location 2000H are B5H, and the contents of memory location 2001H are 18H, then after the execution of

```
RET
```

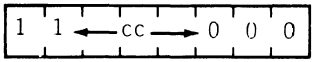
the contents of the Stack Pointer will be 2002H and the contents of the Program Counter will be 18B5H, pointing to the address of the next program opcode to be fetched.

RET cc

Operation: IF cc TRUE: $PC_L \leftarrow (SP)$, $PC_H \leftarrow (SP+1)$

Format:

Opcode	Operands
RET	cc



Description:

If condition cc is true, control is returned to the original program flow by popping the previous contents of the Program Counter (PC) off the top of the external memory stack, where they were pushed by the CALL instruction. This is accomplished by first loading the low-order byte of the PC with the contents of the memory address pointed to by the Stack Pointer (SP), then incrementing the SP, and loading the high-order byte of the PC with the contents of the memory address now pointed to by the SP. (The SP is now incremented a second time.) On the following machine cycle the CPU will fetch the next program opcode from the location in memory now pointed to by the PC. If condition cc is false, the PC is simply incremented as usual, and the program continues with the next sequential instruction. Condition cc is programmed as one of eight status which correspond to condition bits in the Flag Register (register F). These eight status are defined in the table below, which also specifies the corresponding cc bit fields in the assembled object code.

cc	Condition	Relevant Flag
000	NZ non zero	Z
001	Z zero	Z
010	NC non carry	C
011	C carry	C
100	PO parity odd	P/V
101	PE parity even	P/V
110	P sign positive	S
111	M sign negative	S

If cc is true:

M CYCLES: 3 T STATES: 11(5,3,3) 4 MHz E.T.: 2.75

If cc is false:

M CYCLES: 1 T STATES: 5 4 MHz E.T.: 1.25

Condition Bits Affected: None

Example:

If the S flag in the F register is set, the contents of the Program Counter are 3535H, the content of the Stack Pointer are 2000H, the contents of memory location 2000H are B5H, and the contents of memory location 2001H are 18H, then after the execution of

```
RET M
```

the contents of the Stack Pointer will be 2002H and the contents of the Program Counter will be 18B5H, pointing to the address of the next program opcode to be fetched.

RETI

Operation: Return from interrupt

Format:

Opcode

RETI

1	1	1	0	1	1	0	1	ED
0	1	0	0	1	1	0	1	4D

Description:

This instruction is used at the end of an interrupt service routine to:

1. Restore the contents of the Program Counter (PC) (analogous to the RET instruction)

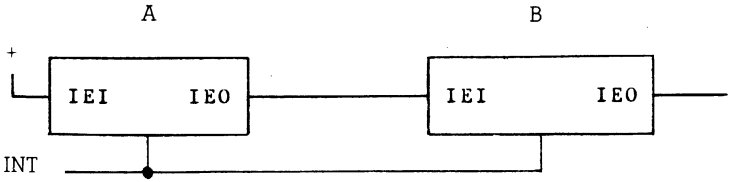
2. To signal an I/O device that the interrupt routine has been completed. The RETI instruction facilitates the nesting of interrupts allowing higher priority devices to suspend service of lower priority service routines. The state of IFF2 is copied into IFF1.

M CYCLES: 4 T STATES: 14(4,4,3,3) 4 MHz E.T.: 3.50

Condition Bits Affected: None

Example:

Given: Two interrupting devices, A and B connected in a daisy chain configuration with A having a higher priority than B.



B generates an interrupt and is acknowledged. (The interrupt enable out, IEO, of B goes low, blocking, any lower priority devices from interrupting while B is being serviced). then A generates an interrupt, suspending service of B. (The IEO of A goes "low" indicating that a higher priority device is being serviced). The A routine is completed and a RETI is issued resetting the IEO of A, allowing the B routine to continue. A second RETI is issued on completion of the B routine and the IEO of B is reset (high) allowing lower priority devices interrupt access.

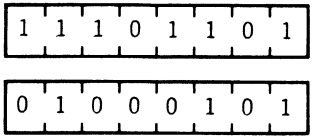
RETN

Operation: Return from non maskable interrupt

Format:

Opcode

RETN



Description:

Used at the end of a service routine for a non maskable interrupt, this instruction executes an unconditional return which functions identical to the RET instruction. That is, the previously stored contents of the Program Counter (PC) are popped off the top of the external memory stack; the low-order byte of PC is loaded with the contents of the memory location pointed to by the Stack Pointer (SP), SP is incremented, the high-order byte of PC is loaded with the contents of the memory location now pointed to by SP, and SP is incremented again. Control is now returned to the original program flow: on the following machine cycle the CPU will fetch the next opcode from the location in memory now pointed to by the PC. Also the state of IFF2 is copied back into IFF1 to the state it had prior to the acceptance of the NMI.

M CYCLES: 4 T STATES: 14(4,4,3,3) 4 MHz E.T.: 3.50

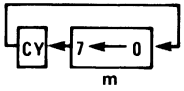
Condition Bits Affected: None

Example:

If the contents of the Stack Pointer are 1000H and the contents of the Program Counter are 1A45H when a non maskable interrupt (NMI) signal is received, the CPU will ignore the next instruction and will instead restart to memory address 0066H. That is, the current Program Counter contents of 1A45H will be pushed onto the external stack address of 0FFFH and 0FFEH, high order-byte first, and 0066H will be loaded onto the Program Counter. That address begins an interrupt service routine which ends with RETN instruction. Upon the execution of RETN, the former Program Counter contents are popped off the external memory stack, low-order first, resulting in a Stack Pointer contents low-order first, resulting in a Stack Pointer contents again of 1000H. The program flow continues where it left off with an opcode fetch to address 1A45H.

RL m

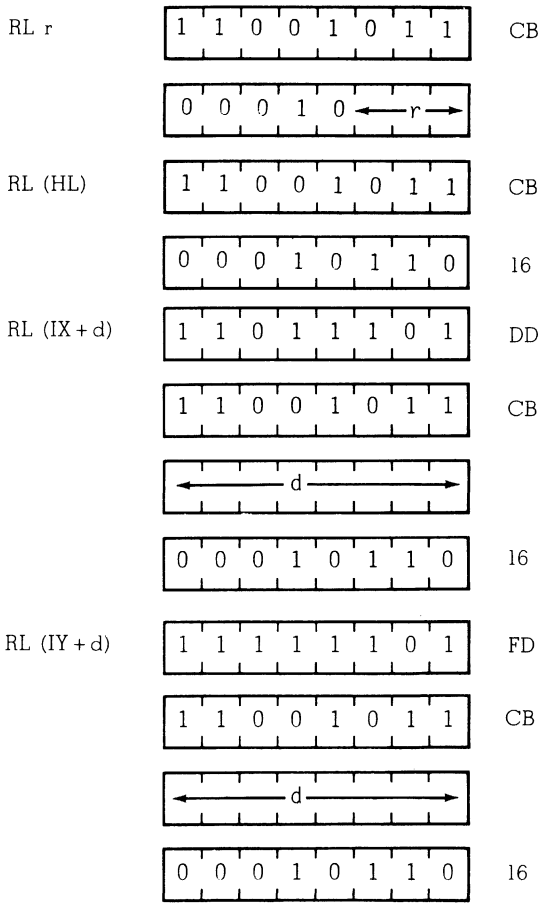
Operation:



Format:

Opcode	Operands
RL	m

The m operand is any of r,(HL), (IX + d) or (IY + d), as defined for the analogous RLC instructions. These various possible opcode-operand combinations are specified as follows in the assembled object code:



r identifies registers B,C,D,E,H,L or A specified as follows in the assembled object code above:

Register	r
B	000
C	001
D	010
E	011
H	011
L	101
A	111

Description:

The contents of the m operand are rotated left: the content of bit 0 is copied into bit 1; the previous content of bit 1 is copied into bit 2; this pattern is continued throughout the byte. The content of bit 7 is copied into the Carry Flag (C flag in register F) and the previous content of the Carry Flag is copied into bit 0 (Bit 0 is the least significant bit.)

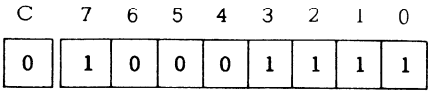
INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
RL r	2	8(4,4)	2.00
RL (HL)	4	15(4,4,4,3)	3.75
RL (IY + d)	6	23(4,4,3,5,4,3)	5.75
RL (IY + d)	6	23(4,4,3,5,4,3)	5.75

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Reset
P/V: Set if parity even; reset otherwise
N: Reset
C: Data from Bit 7 of source register

Example:

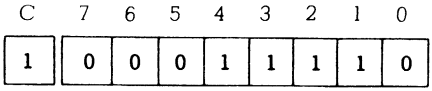
If the contents of register D and the Carry Flag are



after the execution of

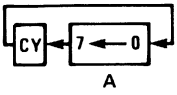
RL D

the contents of register D and the Carry Flag will be



RLA

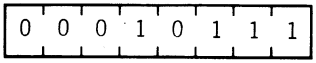
Operation:



Format:

Opcode

RLA



17

Description:

The contents of the Accumulator (register A) are rotated left: the content of bit 0 is copied into bit 1; the previous content of bit 1 is copied into bit 2; this pattern is continued throughout the register. The content of bit 7 is copied into the Carry Flag (C flag in register F) and the previous content of the Carry Flag is copied into bit 0. Bit 0 is the least significant bit.

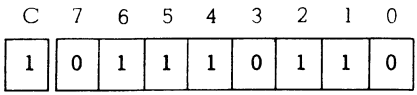
M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Not affected
N: Reset
C: Data from Bit 7 of Acc.

Example:

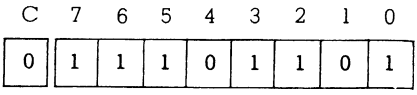
If the contents of the Accumulator and the Carry Flag are



after the execution of

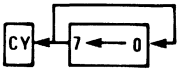
RLA

the contents of the Accumulator and the Carry Flag will be



RLC (HL)

Operation:



Format: (HL)

Opcode	Operands	
RLC	(HL)	
1 1 0 0 1 0 1 1		CB
0 0 0 0 0 1 1 0		06

Description:

The contents of the memory address specified by the contents of register pair HL are rotated left: the content of bit 0 is copied into bit 1; the previous content of bit 1 is copied into bit 2; this pattern is continued throughout the byte. The content of bit 7 is copied into the Carry Flag (C flag in register F) and also into bit 0. Bit 0 is the least significant bit.

M CYCLES: 4 T STATES: 15(4,4,4,3) 4 MHz E.T.: 3.75

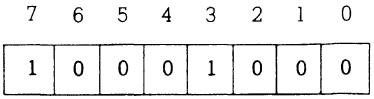
Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Reset
P/V: Set if parity even; reset otherwise
N: Reset
C: Data from Bit 7 of source register

RLC (HL)

Example:

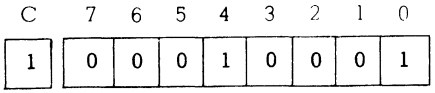
If the contents of the HL register pair are 2828H, and the contents of memory location 2828H are



after the execution of

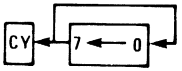
RLC (HL)

the contents of memory location 2828H and the Carry Flag will be



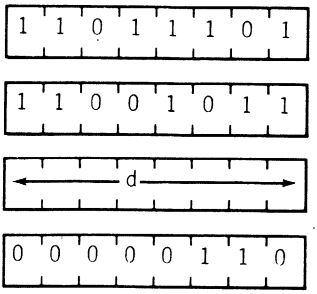
RLC (IX+d)

Operation:



Format:

Opcode	Operands
RLC	(IX + d)



Description:

The contents of the memory address specified by the sum of the contents of the Index Register IX and a two's complement displacement integer d, are rotated left: the contents of bit 0 is copied into bit 1; the previous content of bit 1 is copied into bit 2; this pattern is continued throughout the byte. The content of bit 7 is copied into the Carry Flag (C flag in register F) and also into bit 0. Bit 0 is the least significant bit.

M CYCLES: 6 T STATES: 23(4,4,3,5,4,3) 4 MHz E.T.: 5.75

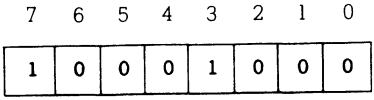
Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Reset
P/V: Set if parity even; reset otherwise
N: Reset
C: Data from Bit 7 of source register

RLC (IX+d)

Example:

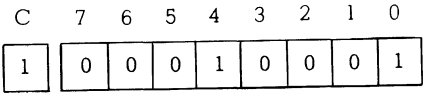
If the contents of the Index Register IX are 1000H, and the contents of memory location 1002H are



after the execution of

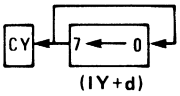
RLC (IX + 2H)

the contents of memory location 1002H and the Carry Flag will be



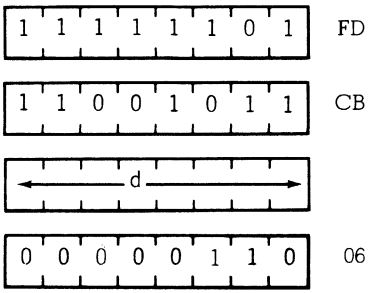
RLC (IY+d)

Operation:



Format:

Opcode	Operands
RLC	(IY + d)



Description:

The contents of the memory address specified by the sum of the contents of the Index Register IY and a two's complement displacement integer d are rotated left: the content of bit 0 is copied into bit 1; the previous content of bit 1 is copied into bit 2; this process is continued throughout the byte. The content of bit 7 is copied into the Carry Flag (C flag in register F) and also into bit 0. Bit 0 is the least significant bit.

M CYCLES: 6 T STATES: 23(4,4,3,5,4,3) 4 MHz E.T.: 5.75

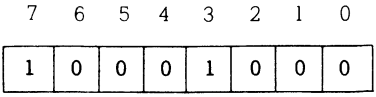
Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Reset
P/V: Set if parity even; reset otherwise
N: Reset
C: Data from Bit 7 of source register

RLC (IY+d)

Example:

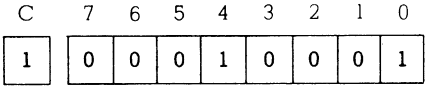
If the contents of the Index Register IY are 1000H, and the contents of memory location 1002H are



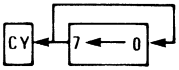
after the execution of

RLC (IY + 2H)

the contents of memory location 1002H and the Carry Flag will be

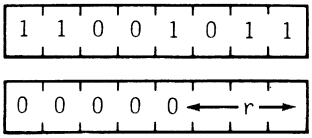


Operation:



Format:

Opcode	Operands
RLC	r



Description:

The eight-bit contents of register r are rotated left: the content of bit 0 is copied into bit 1; the previous content of bit 1 is copied into bit 2; this pattern is continued throughout the register. The content of bit 7 is copied into the Carry Flag (C flag in register F) and also into bit 0. Operand r is specified as follows in the assembled object code:

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Note: Bit 0 is the least significant bit.

M CYCLES: 2 T STATES: (8(4,4) 4 MHz E.T.: 2.00

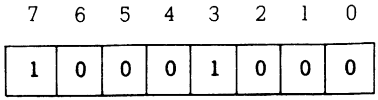
Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Reset
P/V: Set if parity even; reset otherwise
N: Reset
C: Data from Bit 7 of source register

RLC r

Example:

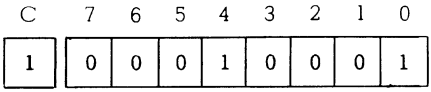
If the contents of register r are



after the execution of

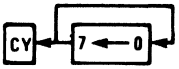
RLC r

the contents of register r and the Carry Flag will be



RLCA

Operation:



Foramt:

Opcode

RLCA

0	0	0	0	0	1	1	1	07

Description:

The contents of the Accumulator (register A) are rotated left: the content of bit 0 is moved to bit 1; the previous content of bit 1 is moved to bit 2; this pattern is continued throughout the register. The content of bit 7 is copied into the Carry Flag (C flag in register F) and also into bit 0. (Bit 0 is the least significant bit).

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Not affected
N: Reset
C: Data from Bit 7 of Acc.

Example:

If the contents of the Accumulator are

7	6	5	4	3	2	1	0
1	0	0	0	1	0	0	0

after the execution of

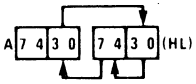
RLCA

the contents of the Accumulator and Carry Flag will be

C	7	6	5	4	3	2	1	0
1	0	0	0	1	0	0	0	1

RLD

Operation:



Format:

Opcode

RLD

1	1	1	0	1	1	0	1	ED
0	1	1	0	1	1	1	1	6F

Description:

The contents of the low order four bits (bits 3,2,1 and 0) of the memory location (HL) are copied into the high order four bits (7,6,5 and 4) of that same memory location; the previous contents of those high order four bits are copied into the low order four bits of the Accumulator (register A); and the previous contents of the low order four bits of the Accumulator are copied into the low order four bits of memory location (HL). The contents of the high order bits of the Accumulator are unaffected. Note: (HL) means the memory location specified by the contents of the HL register pair.

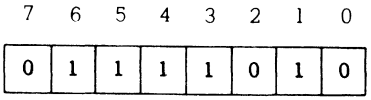
M CYCLES: 5 T STATES: 18(4,4,3,4,3) 4 MHz E.T.: 4.50

Condition Bits Affected:

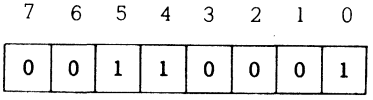
S:	Set if Acc. is negative after operation; reset otherwise
Z:	Set if Acc. is zero after operation; reset otherwise
H:	Reset
P/V:	Set if parity of Acc. is even after operation; reset otherwise
N:	Reset
C:	Not affected

Example:

If the contents of the HL register pair are 5000H, and the contents of the Accumulator and memory location 5000H are



Accumulator

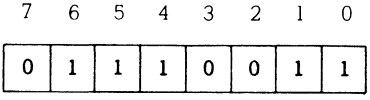


(5000H)

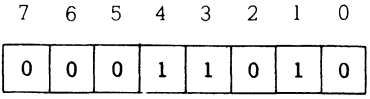
after the execution of

RLD

the contents of the Accumulator and memory location 5000H will be



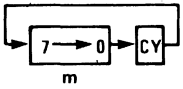
Accumulator



(5000H)

RR m

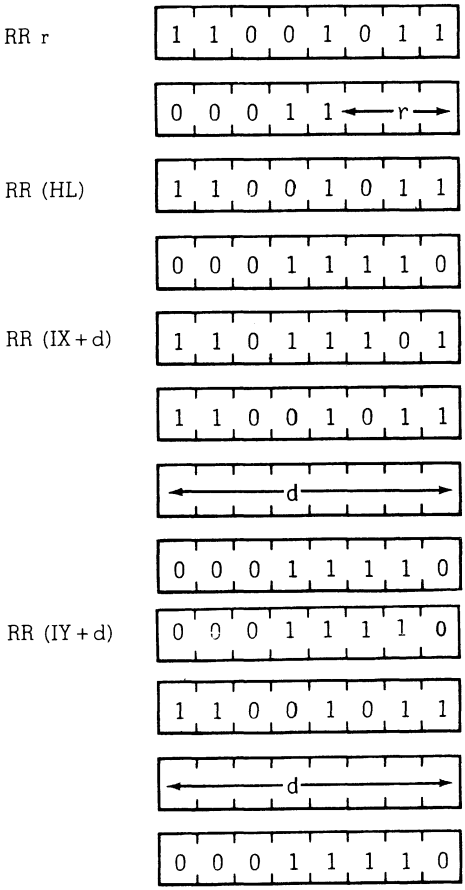
Operation:



Format:

Opcode	Operand
RR	m

The m operand is any of r, (HL), (IX + d), or (IY + d), as defined for the analogous RLC instructions. These various possible opcode-operand combinations are specified as follows in the assembled object code:



r identifies registers B,C,D,E,H,L or A specified as follows in the assembled object code above:

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

The contents of operand m are rotated right: the contents of bit 7 is copied into bit 6; the previous content of bit 6 is copied into bit 5; this pattern is continued throughout the byte. The content of bit 0 is copied into the Carry Flag (C flag in register F) and the previous content of the Carry Flag is copied into bit 7. Bit 0 is the least significant bit.

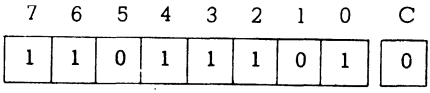
INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
RR r	2	8(4,4)	2.00
RR (HL)	4	15(4,4,4,3)	3.75
RR (IX + d)	6	23(4,4,3,5,4,3)	5.75
RR (IY + d)	6	23(4,4,3,5,4,3)	5.75

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Reset
P/V: Set if parity is even; reset otherwise
N: Reset
C: Data from Bit 0 of source register

Example:

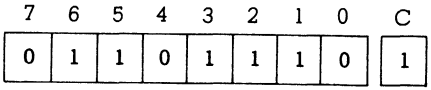
If the contents of the HL register pair are 4343H, and the contents of memory location 4343H and the Carry Flag are



after the execution of

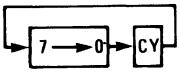
RR (HL)

the contents of location 4343H and the Carry Flag will be



RRA

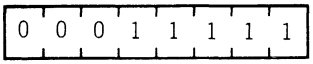
Operation:



Format:

Opcode

RRA



1F

Description:

The contents of the Accumulator (register A) are rotated right: the content of bit 7 is copied into bit 6; the previous content of bit 6 is copied into bit 5; this pattern is continued throughout the register. The content of bit 0 is copied into the Carry Flag (C flag in register F) and the previous content of the Carry Flag is copied into bit 7. Bit 0 is the least significant bit.

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Not affected
N: Reset
C: Data from Bit 0 af Acc.

Example:

If the contents of the Accumulator and the Carry Flag are

7	6	5	4	3	2	1	0	C
1	1	1	0	0	0	0	1	0

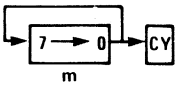
after the execution of

RRA

the contents of the Accumulator and the Carry Flag will be

7	6	5	4	3	2	1	0	C
0	1	1	1	0	0	0	0	1

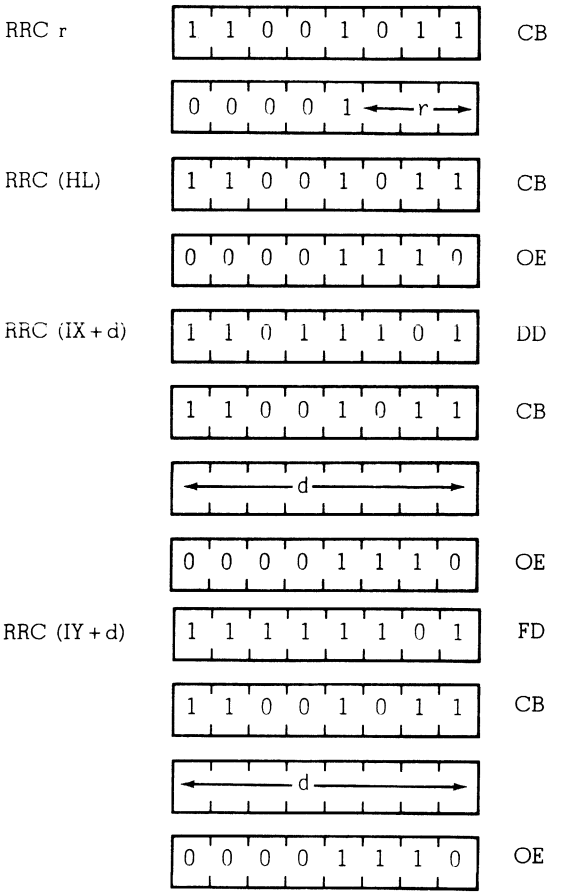
Operation:



Format:

Opcode	Operands
RRC	m

The m operand is any of r,(HL), (IX + d) or (IY + d), as defined for the analogous RLC instructions. These various possible opcode-operand combinations are specified as follows in the assembled object code:



r identifies registers B,C,D,E,H,L or A specified as follows in the assembled object code above:

RRC m

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

The contents of operand m are rotated right: the content of bit 7 is copied into bit 6; the previous content of bit 6 is copied into bit 5; this pattern is continued throughout the byte. The content of bit 0 is copied into the Carry Flag (C flag in the F register) and also into bit 7. Bit 0 is the least significant bit.

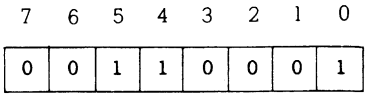
INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
RRC r	2	8(4,4,)	2.00
RRC (HL)	4	15(4,4,4,3)	3.75
RRC (IX + d)	6	23(4,4,3,5,4,3)	5.75
RRC (IY + d)	6	23(4,4,3,5,4,3)	5.75

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Reset
P/V: Set if parity even; reset otherwise
N: Reset
C: Data from Bit 0 af source register

Example:

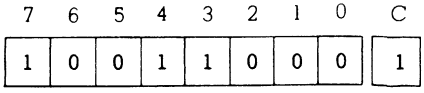
If the contents of register A are



after the execution of

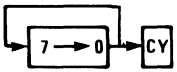
RRC A

the contents of register A and the Carry Flag will be



RRCA

Operation:



Format:

A

Opcode

RRCA

0	0	0	0	1	1	1	1

OF

Description:

The contents of the Accumulator (register A) is rotated right: the content of bit 7 is copied into bit 6; the previous content of bit 6 is copied into bit 5; this pattern is continued throughout the register. The content of bit 0 is copied into bit 7 and also into the Carry Flag (C flag in register F). Bit 0 is the least significant bit.

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Not affected
N: Reset
C: Data from Bit 0 of Acc.

Example:

If the contents of the Accumulator are

7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	1

After the execution of

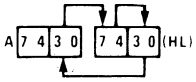
RRCA

the contents of the Accumulator and the Carry Flag will be

7	6	5	4	3	2	1	0	C
1	0	0	0	1	0	0	0	1

RRD

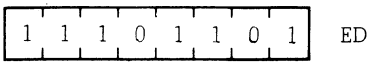
Operation:

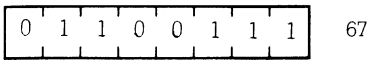


Format:

Opcode

RRD





Description:

The contents of the low order four bits (bits 3,2,1 and 0) of memory location (HL) are copied into the low order four bits of the Accumulator (register A); the previous contents of the low order four bits of the Accumulator are copied into the high order four bits (7,6,5 and 4) of location (HL); and the previous contents of the high order four bits of (HL) are copied into the low order four bits of (HL). The contents of the high order bits of the Accumulator are unaffected. Note: (HL) means the memory location specified by the contents of the HL register pair.

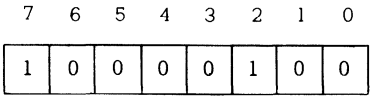
M CYCLES: 5 T STATES: 18(4,4,3,4,3) 4 MHz E.T.: 4.50

Condition Bits Affected:

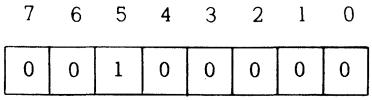
S:	Set if Acc. is negative after operation; reset otherwise
Z:	Set if Acc. is zero after operation; reset otherwise
H:	Reset
P/V:	Set if parity of Acc. is even after operation; reset otherwise
N:	Reset
C:	Not affected

Example:

If the contents of the HL register pair are 5000H, and the contents of the Accumulator and memory location 5000H are



Accumulator

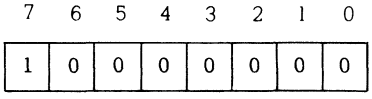


(5000H)

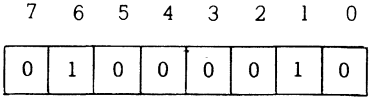
after the execution of

RRD

the contents of the Accumulator and memory location 5000H will be



Accumulator



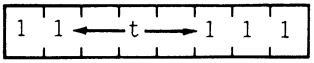
(5000H)

RST p

Operation: (SP-1)←PC_H, (SP-2)←PC_L, PC_H←0, PC_L←P

Format:

Opcode	Operand
RST	p



Description:

The current Program Counter (PC) contents are pushed onto the external memory stack, and the page zero memory location given by operand p is loaded into the PC.
Program execution then begins with the opcode in the address now pointed to by PC. The push is performed by first decrementing the contents of the Stack Pointer (SP), loading the high-order byte of PC into the memory address now pointed to by SP, decrementing SP again, and loading the low-order byte of PC into the address now pointed to by SP. The ReSTart instruction allows for a jump to one of eight addresses as shown in the table below. The operand p is assembled into the object code using the corresponding T state. Note: Since all addresses are in page zero of memory, the high order byte of PC is loaded with 00H. The number selected from the "p" column of the table is loaded into the low-order byte of PC.

p	t
00H	000
08H	001
10H	010
18H	011
20H	100
28H	101
30H	110
38H	111

M CYCLES: 3 T STATES: 11(5,3,3) 4 MHz E.T.: 2.75

Example:

If the contents of the Program Counter are 15B3H, after the execution of

RST 18H (Object code 1101111)

the PC will contain 0018H, as the address of the next opcode to be fetched.

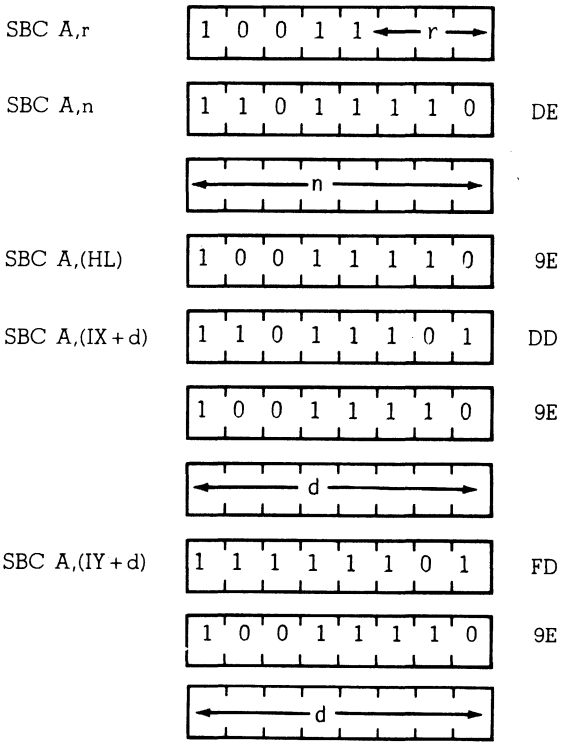
SBC A,s

Operation: A←A - s - CY

Format:

Opcode	Operands
SBC	A,s

The s operand is any of r,n,(HL),(IX + d) or (IY + d) as defined for the analogous ADD instructions. These various possible opcode-operand combinations are assembled as follows in the object code:



r identifies registers B,C,D,E,H,L or A assembled as follows in the object code field above:

SBC A,s

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

The s operand, along with the Carry Flag ("C" in the F register) is subtracted from the contents of the Accumulator, and the result is stored in the Accumulator.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
SBC A,r	1	4	1.00
SBC A,n	2	7(4,3)	1.75
SBC A,(HL)	2	7(4,3)	1.75
SBC A,(IX + d)	5	19(4,4,3,5,3)	4.75
SBC A, (IY + d)	5	19(4,4,3,5,3)	4.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if there is a borrow and reset otherwise.
P/V:	Set if overflow; reset otherwise
N:	Set
C:	Set if there is a borrow and reset otherwise.

Example:

If the Accumulator contains 16H, the carry flag is set, the HL register pair contains 3433H, and address 3433H contains 05H, after the execution of

```
SBC A,(HL)
```

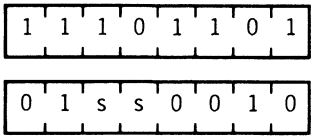
the Accumulator will contain 10H.

SBC HL,ss

Operation: HL←HL-ss-Cy

Format:

Opcode	Operands
SBC	HL,ss



Description:

The contents of the register pair ss (any of register pairs BC,DE,HL or SP) and the Carry Flag (C flag in the F register) are subtracted from the contents of register pair HL and the result is stored in HL. Operand ss is specified as follows in the assembled object code.

Register Pair	ss
BC	00
DE	00
HL	10
SP	11

M CYCLES: 4 T STATES: 15(4,4,4,3) 4 MHz E.T.: 3.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set if there is a borrow and reset otherwise.
P/V:	Set if overflow; reset otherwise
N:	Set
C:	Set if there is a borrow and reset otherwise.

Example:

If the contents of the HL register pair are 9999H, the contents of register pair DE are 1111H, and the Carry Flag is set, after the execution of

```
SBC HL,DE
```

the contents of HL will be 8887H.

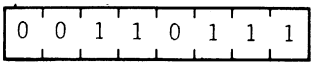
SCF

Operation: CY←1

Format:

Opcode

SCF



37

Description:

The C flag in the F register is set.

M CYCLES: 1 T STATES: 4 4 MHz E.T.: 1.00

Condition Bits Affected:

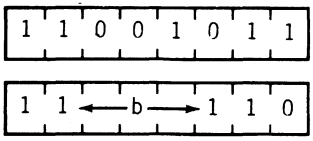
S: Not affected
Z: Not affected
H: Reset
P/V: Not affected
N: Reset
C: Set

SET b,(HL)

Operation: $(HL)_b \leftarrow 1$

Format:

Opcode	Operands
SET	b,(HL)



Description:

Bit b (any bit, 7 through 0) in the memory location addressed by the contents of register pair HL is set. Operand b is specified as follows in the assembled object code:

Bit Tested	b
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

M CYCLES: 4 T STATES: 15(4,4,4,3) 4 MHz E.T.: 3.75

Condition Bits Affected: None

Example:

If the contents of the HL register pair are 3000H, after the execution of

```
SET 4,(HL)
```

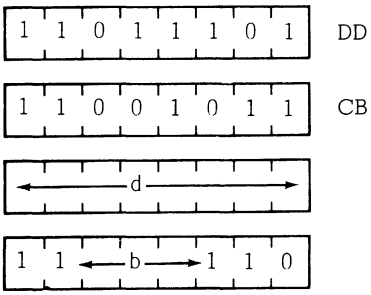
bit 4 in memory location 3000H will be 1. (Bit 0 in memory location 3000H is the least significant bit.)

SET b,(IX+d)

Operation: $(IX+d)_b \leftarrow 1$

Format:

Opcode	Operands
SET	b,(IX + d)



Description:

Bit b (any bit, 7 through 0) in the memory location addressed by the sum of the contents of the IX register pair (Index Register IX) and the two's complement integer d is set. Operand d is specified as follows in the assembled object code:

Bit Tested	b
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

M CYCLES: 6 T STATES: 23(4,4,3,5,4,3) 4 MHz E.T.: 5.75

Condition Bits Affected: None

Example:

If the contents of Index Register are 2000H, after the execution of

SET 0,(IX + 3H)

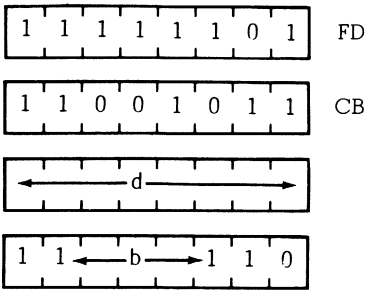
bit 0 in memory location 2003H will be 1. (Bit 0 in memory location 2003H is the least significant bit.)

SET b,(IY+d)

Operation: $(IY + d)_b \leftarrow 1$

Format:

Opcode	Operands
SET	b,(IY + d)



Description:

Bit b (any bit, 7 through 0) in the memory location addressed by the sum of the contents of the IY register pair (Index Register IY) and the two's complement displacement d is set. Operand b is specified as follows in the assembled object code:

Bit Tested	b
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

M CYCLES: 6 T STATES: 23(4,4,3,5,4,3) 4 MHz E.T.: 5.75

Condition Bits Affected: None

Example:

If the contents of Index Register IY are 2000H, after the execution of

SET 0,(IY + 3H)

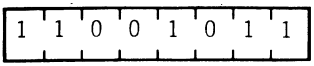
bit 0 in memory location 2003H will be 1. (Bit 0 in memory location 2003H is the least significant bit.)

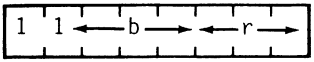
SET b,r

Operation: $r_b \leftarrow 1$

Format:

Opcode	Operands
SET	b,r





Description:

Bit b (any bit, 7 through 0) in register r (any of registers B,C,D,E,H,L or A) is set. Operands b and r are specified as follows in the assembled object code:

Bit	b	Register	r
0	000	B	000
1	001	C	001
2	010	D	010
3	011	E	011
4	100	H	100
5	101	L	101
6	110	A	111
7	111		

M CYCLES: 2 T STATES: 8(4,4) 4 MHz E.T.: 2.00

Condition Bits Affected: None

Example:

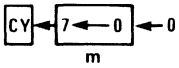
After the execution of

```
SET 4,A
```

bit 4 in register A will be set. (Bit 0 is the least significant bit.)

SLA m

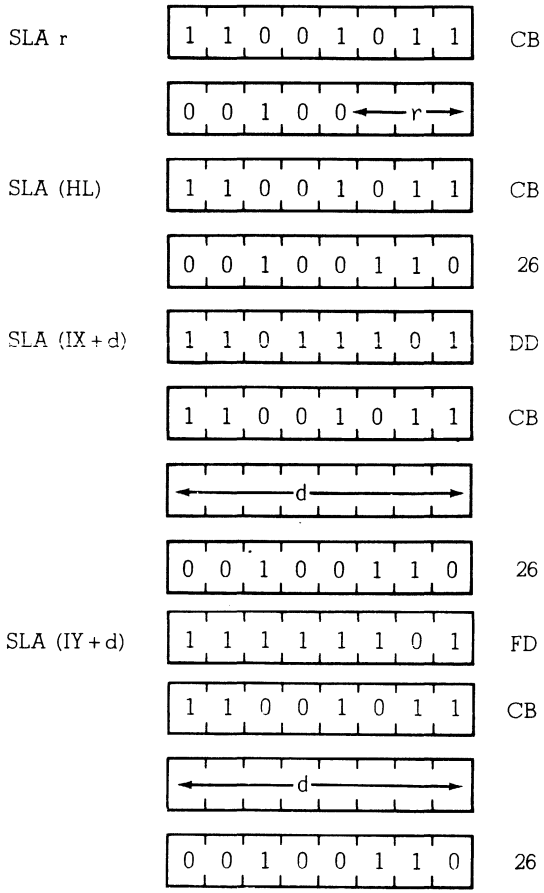
Operation:



Format:

Opcode	Operands
SLA	m

The m operand is any of r, (HL), (IX + d) or (IY + d), as defined for the analogous RLC instructions. These various possible opcode-operand combinations are specified as follows in the asembled object code:



r identifies registers B,C,D,E,H,L or A specified as follows in the assembled object code field above:

SLA m

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

An arithmetic shift left is performed on the contents of operand m: bit 0 is reset, the previous content of bit 0 is copied into bit 1, the previous content of bit 1 is copied into bit 2; this pattern is continued throughout; the content of bit 7 is copied into the Carry Flag (C flag in register F). Bit 0 is the least significant bit.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
SLA r	2	8(4,4)	2.00
SLA (HL)	4	15(4,4,4,3)	3.75
SLA (IX + d)	6	23(4,4,3,5,4,3)	5.75
SLA (IY + d)	6	23(4,4,3,5,4,3)	5.75

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Reset
P/V: Set if parity is even; reset otherwise
N: Reset
C: Data from Bit 7

Example:

If the contents of register L are

7	6	5	4	3	2	1	0
1	0	1	1	0	0	0	1

after the execution of

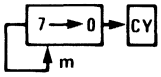
SLA L

the contents of register L and the Carry Flag will be

C	7	6	5	4	3	2	1	0
1	0	1	1	0	0	0	1	0

SRA m

Operation:



Format:

Opcode	Operands
SRA	m

The m operand is any of r, (HL), (IX + d) or (IY + d), as defined for the analogous RLC instructions. These various possible opcode-operand combinations are specified as follows in the assembled object code:

SRA r	1 1 0 0 1 0 1 1	CB
	0 0 1 0 1 ← r →	
SRA(HL)	1 1 0 0 1 0 1 1	CB
	0 0 1 0 1 1 1 0	2E
SRA(IX + d)	1 1 0 1 1 1 0 1	DD
	1 1 0 0 1 0 1 1	CB
	← d →	
	0 0 1 0 1 1 1 0	2E
SRA(IY + d)	1 1 1 1 1 1 0 1	FD
	1 1 0 0 1 0 1 1	CB
	← d →	
	0 0 1 0 1 1 1 0	2E

* r means registers B,C,D,E,H,L or A specified as follows in the assembled object code field above:

SRA m

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

An arithmetic shift right is performed on the contents of operand m: the content of bit 7 is copied into bit 6; the previous content of bit 6 is copied into bit 5; this pattern is continued throughout the byte. The content of bit 0 is copied into the Carry Flag (C flag in register F), and the previous content of bit 7 is unchanged. Bit 0 is the least significant bit.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
SRA r	2	8(4,4)	2.00
SRA (HL)	4	15(4,4,4,3)	3.75
SRA (IX + d)	6	23(4,4,3,5,4,3)	5.75
SRA (IY + d)	6	23(4,4,3,5,4,3)	5.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Reset
P/V:	Set if parity is even; reset otherwise
N:	Reset
C:	Data from Bit 0 of source register

Example:

If the contents of the Index Register IX are 1000H, and the contents of memory location 1003H are

7	6	5	4	3	2	1	0
1	0	1	1	1	0	0	0

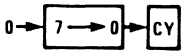
after the execution of

SRA (IX + 3H)

the contents of memory location 1003H and the Carry Flag will be

7	6	5	4	3	2	1	0	C
1	1	0	1	1	1	0	0	0

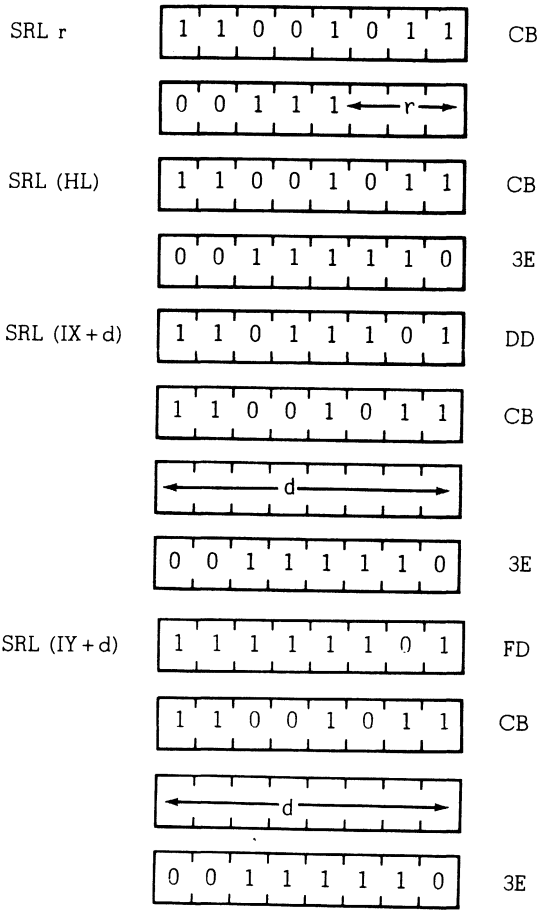
Operation:



Format:

Opcode	Operands
SRL	m

The operand m is any of r, (HL), (IX + d) or (IY + d), as defined for the analogous RLC instructions. These various possible opcode-operand combinations are specified as follows in the assembled object code:



r identifies registers B,C,D,E,H,L or A specified as follows in the assembled object code fields above:

SRL m

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

The contents of operand m are shifted right: the content of bit 7 is copied into bit 6; the content of bit 6 is copied into bit 5; this pattern is continued throughout the byte. The content of bit 0 is copied into the Carry Flag, and bit 7 is reset. Bit 0 is the least significant bit.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
SRL r	2	8(4,4)	2.00
SRL (HL)	4	15(4,4,4,3)	3.75
SRL (IY + d)	6	23(4,4,3,5,4,3)	5.75
SRL (IY + d)	6	23(4,4,3,5,4,3)	5.75

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Reset
P/V: Set if parity is even; reset otherwise
N: Reset
C: Data from Bit 0 of source register

Example:

If the contents of register B are

7	6	5	4	3	2	1	0
1	0	0	0	1	1	1	1

after the execution of

SRL B

the contents of register B and the Carry Flag will be

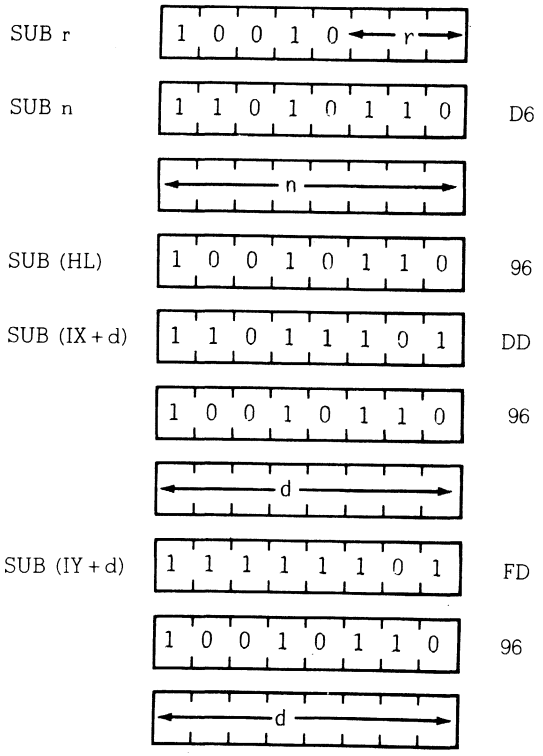
7	6	5	4	3	2	1	0	C
0	1	0	0	0	1	1	1	1

Operation: A←A - s

Format:

Opcode	Operands
SUB	s

The s operand is any of r,n,(HL), (IX + d) or (IY + d) as defined for the analogous ADD instruction. These various possible opcode-operand combinations are assembled as follows in the object code:



r identifies registers B,C,D,E,H,L or A assembled as follows in the object code field above:

SUB s

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

The s operand is subtracted from the contents of the Accumulator, and the result is stored in the Accumulator.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
SUB r	1	4	1.00
SUB n	2	7(4,3)	1.75
SUB (HL)	2	7(4,3)	1.75
SUB (IX + d)	5	19(4,4,3,5,3)	4.75
SUB (IY + d)	5	19(4,4,3,5,3)	4.75

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Set if there is a borrow and reset otherwise
P/V: Set if overflows; reset otherwise
N: Set
C: Set if there is a borrow and reset otherwise.

Example:

If the Accumulator contains 29H and register D contains 11H, after the execution of

```
SUB D
```

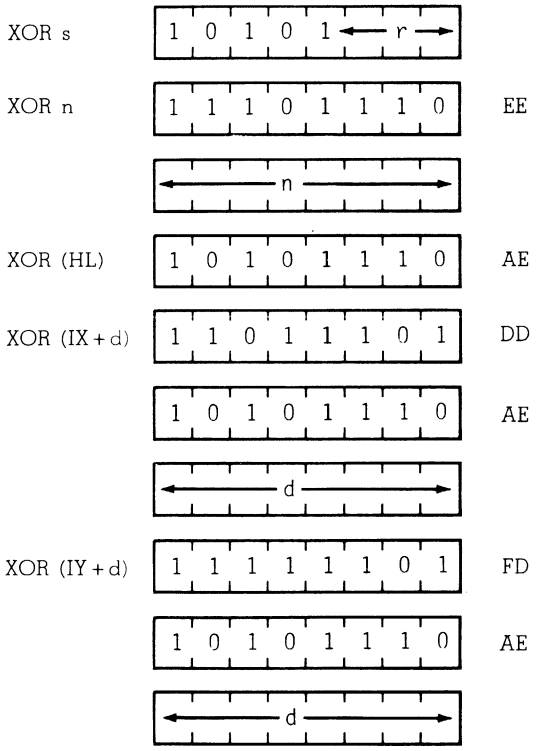
the Accumulator will contain 18H.

Operation: A←A⊕s

Format:

Opcode	Operands
XOR	s

The s operand is any of r,n,(HL), (IX + d) or (IY + d), as defined for the analogous ADD instructions. These various possible opcode-operand combinations are assembled as follows in the object code:



r identifies registers B,C,D,E,H,L or A assembled as follows in the object code field above:

XOR s

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Description:

A logical exclusive-OR operation, bit by bit, is performed between the byte specified by the s operand and the byte contained in the Accumulator; the result is stored in the Accumulator.

INSTRUCTION	M CYCLES	T STATES	4 MHz E.T.
XOR r	1	4	1.00
XOR n	2	7(4,3)	1.75
XOR (HL)	2	7(4,3)	1.75
XOR (IX + d)	5	19(4,4,3,5,3)	4.75
XOR (IY + d)	5	19(4,4,3,5,3)	4.75

Condition Bits Affected:

S:	Set if result is negative; reset otherwise
Z:	Set if result is zero; reset otherwise
H:	Set
P/V:	Set if parity even; reset otherwise
N:	Reset
C:	Reset

Example:

If the Accumulator contains 96H (10010110), after the execution of

XOR 5DH (Note: 5DH = 01011101)

the Accumulator will contain CBH (11001011).

SGS-ATES GROUP OF COMPANIES

Italy - Brazil - France - Malta - Malaysia - Singapore - Sweden - Switzerland - United Kingdom - U.S.A. - West Germany

SGS-ATES GROUP OF COMPANIES

INTERNATIONAL HEADQUARTERS
SGS-ATES Componenti Elettronici SpA
Via C. Olivetti 2, - 20041 Agrate Brianza-Italy
Tel.: 39 - 65551
Telex: 330131 - 330141 - SGSAGR

BENELUX
SGS-ATES Componenti Elettronici SpA
Sales Office:
B- 1180 Bruxelles
Winston Churchill Avenue, 122
Tel.: 2 - 3432439
Telex: 24149 B

BRÃZIL
SGS Semicondutores LTDA
Sales Office:
05413 Sao Paulo
Av. Henrique Schaumann 286 - CJ33
Tel.: 11 - 853-5062
Telex: 37988 UMBR BR

DENMARK
SGS Semicondutor A.B.
Sales Office:
2730 Herlev
Herlev Torv, 4
Tel.: 2 - 948533
Telex: 35411

FRANCE
Société Générale de Semiconducteurs
92120 Montrouge
21-23 Rue de la Vanne
Tel.: 1 - 6571133
Telex: 250938F

HONG KONG
SGS Semiconductour Asia Limited
9th Floor, Block N,
Kaiser Estate, Phase III,
11 Hok Yuen St.,
Hunghom, Kowloon
Tel.: 3-644251/5
Telex: HX 63906 ESGIE HX

ITALY
SGS-ATES Componenti Elettronici SpA
Direzione Italia e Sud Europa
20090 Assago (MI)
V.le Milanofiori - Strada 4 - Palazzo A/4/A
Tel.: 2 - 8244131 (10 linee)
Telex: 330131 - 330141 SGSAGR

Sales Offices:
40128 Bologna
Via Corticella, 231
Tel.: 51-324486
00161 Roma
Via A. Torlonia, 15
Tel.: 6-8444474

SINGAPORE
SGS Semiconductor (Pte) Ltd.
Singapore 2056
28 Ang Mo Kio
Industrial Park 2
tel.: 482-1411
Telex: RS 21412 ESGIES

SWEDEN
SGS Semiconductor A.B.
19500 Märsta
Bristagatan, 16
Tel.: 760 - 40120
Telex: 042 - 10932

SWITZERLAND
SGS Semiconductor S.A.
Sales Offices:
1218 Grand-Saconnex (Geneve)
Chemin François-Lehmann, 22
Tel.: 22 - 986462/3
Telex: 28895
6340 Baar
Oberneuhofstrasse, 2
Tel.: 42 - 315955
Telex: 864915

TAIWAN-REPUBLIC OF CHINA
SGS-ATES Singapore (Pte) Ltd
Taipei Sec 4
6th floor, Pacific Commercial Bldg.
285 Chung Hsiao E Road
Tel.: 2-7728203
Telex: 10310 ESGIE TWN

UNITED KINGDOM
SGS Semiconductor Limited
Aylesbury, Bucks
Planar House, Walton Street
Tel.: 296 - 5977
Telex: 041 - 83245

WEST GERMANY
SGS Halbleiter Bauelemente GmbH
8018 Grafing bei München
Haidling, 17
Tel.: 8092-690
Telex: 05 27378

Sales Offices:
3012 Langenhagen
Hans Boeckler Str., 2
Tel.: 511 - 789881
Telex: 923195
8500 Nürnberg 40
Allersberger Str., 95
eingang Wilhelminenstr. 1
Tel.: 911 - 464071
Telex: 626243
8023 Pullach bei München
Seitnerstrasse, 42
Tel.: 89 - 793 0662
Telex: 5215784
7000 Stuttgart 80
Kalifenweg, 45
Tel.: 711 - 713091/2
Telex: 07 255545

U.S.A.
SGS-Semiconductor Corporation
Phoenix, AZ 85022
1000 East Bell Road
Tel.: (602) 867-6100
Telex: 249976 SGSPH UR

Sales Offices:
Austin, TX 78723
6448 Highway 290 East
Suite F-103
Tel.: (512) 459-4545
Bloomington, MN 55438
6100 Green Valley Drive
Suite 120
Tel.: (612) 835-1347
Telex: 201692 SGSMI UR
Dallas, TX 75248
16970, Dallas North Parkway
Suite 401
Tel.: (214) 733-1515
Telex: 203997 SGSDA UR
Hauppauge, NY 11788
330 Motor Parkway
Suite 100
Tel.: (516) 435-1050
Telex: 221275 SGSHA UR
Indianapolis, IN 46241
2346 S. Lynhurst Drive
Suite H-200
Tel.: (317) 241-1116
Telex: 209144 SGSIN UR
Irvine, CA 92714
18271 W. McDurmott Drive
Suite J.
Tel. (714) 863-1222
Telex: 277793 SGSOR UR
Norcross, GA 30071
3040-D1 Holcomb Bridge Road
Tel.: (404) 446-8686
Telex: 261395 SGSAT UR
Poughkeepsie, NY 12601
201 South Avenue
Suite 206
Tel.: (914) 473-2255
Santa Clara, CA 95051
2700 Augustine Drive
Suite 209
Tel.: (408) 727-3404
Telex: 278833 SGSSA UR
Schaumburg, IL 60196
600 North Meacham Road
Tel.: (312) 490-1890
Telex: 210159 SGSCH UR
Waltham, MA 02154
240 Bear Hill Road
Tel.: (617) 890-6688
Telex: 200297 SGSWH UR
Westlake Village, CA 91361
31416 West Agoura Road
Suite 200
Tel.: (818) 991-2900
Telex: 215258 SGSWD UR
Woodland Hills, CA 91367
6355 Topanga Canyon Boulevard
Suite 220
Tel.: (213) 716-6600
Telex: 215258 SGSWD UR