CHAPTER 13
Z80185/195 INSTRUCTION SET

13.1 INTRODUCTION

This chapter describes the instruction set of the Z80185/195 family processors. To minimize the number of pages required and to eliminate redundant information, instructions that differ only in where the operand(s) reside are described together. After some introductory sections, all instructions are described in alphabetical order, typically in one page per instruction. Each description includes the following parts:

3. The header line includes the assembly language mnemonic and operand(s). These items identify the instruction. So that one instruction description can cover many different instructions, operands are often given as an alphabetic code that stands for many possible values. Such codes are listed in the next section.

4. The Operation line presents the function of the instruction in a symbolic form.

5. The Format section shows how the instruction(s) is (are) structured in memory. For instruction types with more than one kind of operand, separate formats are given for each kind.

6. The Description presents the function of the instruction in text form.

7. Timing shows how many bus cycles (M Cycles) and minimum number of clocks (T States) the processor requires to execute the instruction. For instructions with more than one kind of operand, separate timings may be given for each kind.

   In this manual, the number of M Cycles reflects only the number of read and write bus cycles that occur during the fetching and execution of the instruction. Internal idle times are not included in the number of M Cycles. The minimum number of T States (clocks) required to execute each instruction are given separately for the Z80 and Z8018x families, each total being followed by a breakdown of the number of clocks required for each M Cycle and internal idle period (the latter are shown as "n idle") in the instruction. As both the Z80185 and Z80195 utilize the Z8S180 core, the Z8018x timings are appropriate. The Z80 timings are included for reference only. In some Z80 T State breakdowns, short internal idle times may not be broken out from the M Cycle which they precede or follow. For I/O instructions of the Z80180x family including the Z80185/195, T States are given separately for registers in the "180 core" (those which are typically addressed 0-3F) and for all other on-chip and off-chip registers, reflecting the fact that the processor automatically inserts one "wait state" in M Cycles for registers in the latter category.

8. Condition Bits Affected shows whether and how the instruction affects the Flag register F.

9. An example is included for almost all instruction types.

13.2 OPERAND CODES

(aa) (mn), (IX+d), (IY+d), or the 8-bit contents of memory at the address in BC, DE, or HL.

cc a condition code Z, NZ, C, NC, PE, PO, P, or M

cc' a condition code Z, NZ, C, or NC

d a signed 8-bit displacement, from an index register or the Program Counter PC

ee a 16-bit register BC, DE, HL, SP, IX, or IY.

(IX+d), (IY +D) The 8-bit contents of memory at the address obtained by adding the contents of index register IX or IY and the signed 8-bit displacement d given in the instruction.

m r or (HL) or (IX+d) or (IY+d)

mn a 16-bit "immediate" or "literal" address or value given in the instruction

(mn) the 8-bit contents of memory at an address given in the instruction

n an 8-bit "immediate" or "literal" value given in the instruction

pp a 16-bit register BC, DE, HL, IX, IY, or the concatenation of the Accumulator A as the MSbyte and the flags F as the LSbyte

qq a 16-bit register BC, DE, HL, or the concatenation of the Accumulator A as the MSbyte and the flags F as the LSbyte

r an 8-bit register A, B, C, D, E, H, or L.

rr a 16-bit register HL, IX, or IY

s r or n or (HL) or (IX+d) or (IY+d)

ss a 16-bit register BC, DE, HL, or the Stack Pointer SP.

tt like ss, except that the encoded value that corresponds to HL in ss means "the destination register HL, IX, or IY".

13.3 Z80 STATUS INDICATORS (FLAGS)

The flag registers (F and $F^{|}$) supply information to the user regarding the statue of the Z80 at any given time. The bit positions for each flag is shown below:

7	6	5	4	3	2	1	0
S	Z	X	N	X	P/V	N	C

where:

C	=	Carry Flag
N	=	Add/Subtract
P/V	=	Parity/Overflow Flag
H	=	Half Carry Flag
Z	=	Zero Flag
S	=	Sign Flag
X	=	Not Used

Each of the two Z80 Flag Registers contains six bits of statue information which are set or reset by CPU operations. (Bits 3 and 5 are not used.) Four of these bits are testable (C, P/V, Z, and S) for use with conditional Jump, Call, or Return instructions. Two flags are not testable (H, N) and are used for BCD arithmetic.

13.3.1 Carry Flag (C)

The carry bit is set or reset depending on the operation being performed. For 'ADD' instructions that generate a carry and 'SUBTRACT' instructions that generate a borrow, the Carry flag will be set. The Carry flag is reset by an ADD that does not generate a carry and by a 'SUBTRACT' that generates no borrow. This saved carry facilitates software routines for extended precision arithmetic. Also, the 'DAA' instruction will set the Carry flag if the conditions for making the decimal adjustment are met.

For instructions RLA, RRA, RL, and RR, the carry bit is used as a link between the LSB and MSB for any register or memory location. During instructions RLCA, RLC, and SLA, the carry contains the last value shifted out of bit 7 of any register or memory location. During instructions RRCA, RRC, SRA, and SRL the carry contains the last value shifted out of bit 0 of any register or memory location.

For the logical instructions AND, OR, and XOR, the carry will be reset.

The Carry flag can also be set (SCF) and complemented (CCF).

13.3.2 Add/Subtract Flag (N)

This flag is used by the decimal adjust accumulator instruction (DAA) to distinguish between 'ADD' and 'SUBTRACT' instructions. For all 'ADD' instructions, N will be set to a '0'. For all 'SUBTRACT' instructions, N will be set to a '1.'

13.3.3 Parity/Overflow Flag (P/V)

This flag is set to a particular state depending on the operation being performed.

For arithmetic operations, this flag indicates an overflow condition when the result in the Accumulator is greater than the maximum possible number (+127) or is less than the minimum possible number (–128). This overflow condition can be determined by examining the sign bits of the operands.

For addition, operands with different signs will never cause overflow. When adding operands with like signs and the result has a different sign, the overflow flag is set. For example:

+120	=	0111	1000		
+105	=	0110	1001		
+225	=	1110	0001	(–95)	SUM

The two numbers added together have resulted in a number that exceeds +127 and the two positive operands has resulted in a negative number (–95) which is incorrect. The Overflow flag is therefore set.

For subtraction, overflow can occur for operands of unlike signs. Operands of like sign will never cause overflow. For example:

	+127	0111	1111	MINUEND
(–)	–64	1100	0000	SUBTRAHEND
	+191	1011	1111	DIFFERENCE

The minuend sign has changed from a positive to a negative, giving an incorrect difference. Overflow is therefore set.

Another method for predicting an overflow is to observe the carry into and out of the sign bit. If there is a carry in and no carry out, or if there is no carry in and a carry out, then overflow has occurred.

This flag is also used with logical operations and rotate instructions to indicate the parity of the result. The number of '1' bits in a byte are counted. If the total is odd, 'ODD' parity (P=0) is flagged. If the total is even, 'EVEN' parity is flagged (P=1).

During search instructions (CPI, CPIR, CPD, CPDR) and-block transfer instructions (LDI, LDIR, LDD, LDDR) the P/V flag monitors the state of the byte count register (BC). When decrementing, if the byte counter results in a zero value, the flag is reset to 0, otherwise the flag is set to1.

During LD A, I and LD A, R instructions, the P/V flag is set with the contents of the interrupt enable flip-flop (IFF2) for storage or testing.

When inputting a byte from an I/O device, IN r, (C), the P/V flag is adjusted to indicate the parity of the data.

13.3.4 Half Carry Flag (H)

The Half Carry Flag (H) is set or reset depending on the carry and borrow status between bits 3 and 4 of an 8-bit arithmetic operation. This flag is used by the decimal adjust accumulator instruction (DAA) to correct the result of a packed BCD add or subtract operation. The H flag is set (1) or reset (0) according to the following table:

H	Add	Subtract
1	There is a carry from Bit 3 to Bit 4	There is a borrow from Bit 4
0	There is no carry from Bit 3 to Bit 4	There is a borrow from Bit 4

13.3.5 Zero Flag (Z)

The Zero Flag (Z) is set or reset if the result generated by the execution of certain instructions is zero.

For 8-bit arithmetic and logical operations, the Z flag is set to '1' if the resulting byte in the Accumulator is zero. If the byte is not zero, the Z flag is reset to '0'.

For compare (search) instructions, the Z flag is set to '1' if the value in the Accumulator is equal to the value in the memory location pointed to by the contents of the register pair HL.

When testing a bit in a register or memory location, the Z flag contains the complemented state of the indicated bit (see Bit b, s).

When inputting or outputting a byte between a memory location and an I/O device (INI, IND, OUTI, and OUTD), if the result of decrementing the B register is zero, the Z flag is set, otherwise it is reset. Also for byte inputs from I/O devices using IN r, (C), the Z Flag is set to indicate a zero byte input.

13.3.6 Sign Flag(S)

The Sign Flag (S) stores the state of the most significant bit of the Accumulator (bit 7). When the Z80 performs arithmetic operations on signed numbers, binary two's complement notation is used to represent and process numeric information. A positive number is identified by a '0' in bit 7. A negative number is identified by a '1.' The binary equivalent of the magnitude of a positive number is stored in bits 0 to 6 for a total range of from 0 to 127. A negative number is represented by the two's complement of the equivalent positive number. The total range for negative numbers is from –1 to –128.

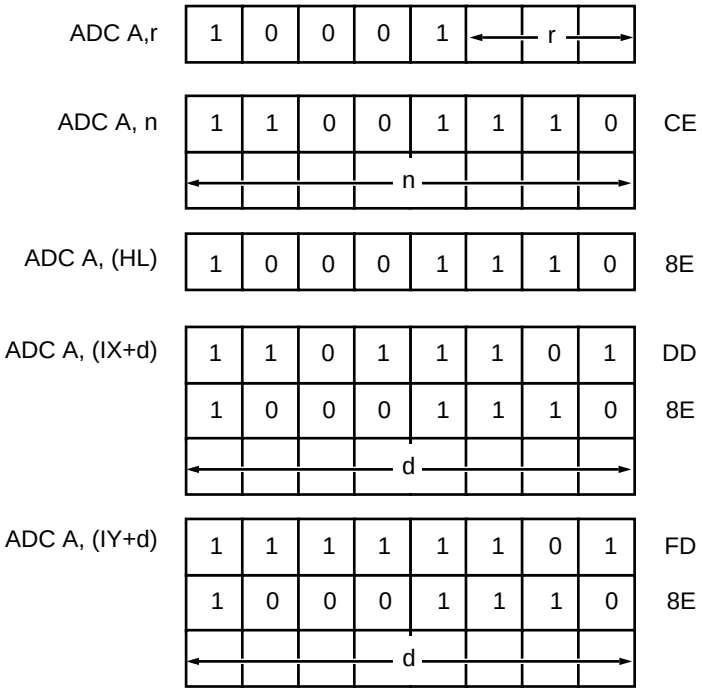
When inputting a byte from an I/O device to a register, IN r, (C), the S flag will indicate either positive (S=0) or negative (S=1) data.

13.4 THE INSTRUCTION SET

ADC A, s

Operation: A ← A + s + CY

Format:



Description: The s operand and the Carry Flag ("C" in the F register) are added to the contents of the Accumulator A, the result is stored in A, and the flags are set as described below. "s" can be any of a register r, an "immediate" value n in the instruction itself, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the source register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
ADC A, r	1	4	4 (3+ 1 int)
ADC A, n	2	7 (4, 3)	6 (3, 3)
ADC A, (HL)	2	7 (4, 3)	6 (3, 3)
ADC A, (IX+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)
ADC A, (IY+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)

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 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.

ADC HL, ss

Operation: HL ← HL + ss + CY

Format:

1	1	1	0	1	1	0	1	ED
0	1	s	s	1	0	1	0	

Description: The contents of register pair ss (any of register pairs BC, DE, HL, or SP) and the Carry flag (C flag in the F register) are added to the contents of register pair HL, the result is stored in HL, and the C flag reflects whether there was a carry out. Operand ss is specified as follows in the assembled object code:

Register Pair	ss
BC	00
DE	01
HL	10
SP	11

Timing:

M Cycles	Z80 T States	Z18x T States
2	15 (4, 4, 7 int)	10 (3, 3, 4 int)

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
R: 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

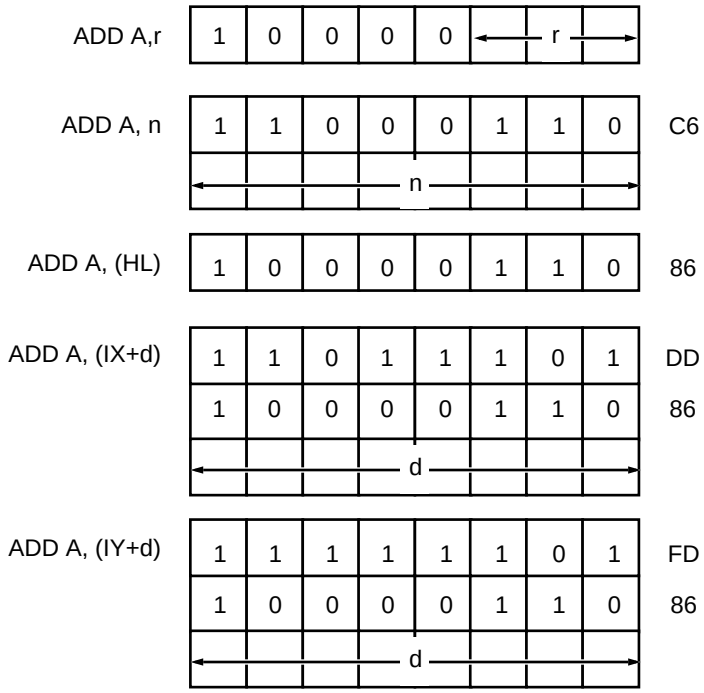
```
ADC HL, BC
```

the contents of HL will be 765AH.

ADD, A, s

Operation: A ← A + s

Format:



Description: The s operand is added to the value in the Accumulator (A), the result is stored in A, and the flags are set as described below. "s" can be any of a register r, an "immediate" value n in the instruction itself, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects a source register as follows:

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

: : Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
ADD A,r	1	4	4 (3, 1 int)
ADD A,n	2	7 (4, 3)	6 (3, 3)
ADD A,(HL)	2	7 (4, 3)	6 (3, 3)
ADD A,(IX+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)
ADD A,(IY+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)

Condition Bits Affected:

S: Set if result is negative, reset otherwise
Z: Set if result is 0; 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 (A) contains 11H, the Index Register IY contains 1000H, and memory location 1005H contains 22H, then after the execution of

ADD A, (IY + 5)

A will contain 33H, and the S, Z, H, P/V, and C flags will all be reset.

ADD rr, tt

Operation: rr ← rr + tt

Format:

ADD HL, tt
(tt = BC, DC, HL, or SP)

0	0	t	t	1	0	0	1

ADD IX, tt
(tt = BC, DE, IX, or SP)

1	1	0	1	1	1	0	1	DD
0	0	t	t	1	0	0	1	

ADD IY, tt
(tt = BC, DE, IY, or SP)

1	1	1	1	1	1	0	1	FD
0	0	t	t	1	0	0	1	

Description: The contents of 16-bit register tt are added to the contents of 16-bit register rr, the result is stored in rr, and the C flag reflects whether there was a carry out. The destination register "rr" can be register pair HL or an index register IX or IY. The source register "tt" is encoded as follows:

Register	tt
BC	00
DE	01
same as rr (HL, IX, or IY)	10
SP	11

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
ADD HL, tt	1	11 (4, 7 int)	7 (3, 4 int)
ADD IX, tt	2	15 (4, 4, 7 int)	10 (3, 3, 4 int)
ADD IY, tt	2	15 (4, 4, 7 int)	10 (3, 3, 4 int)

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 out of bit 15; reset otherwise

Example: If BC contains 0102H and IX contains 1234H, then after execution of

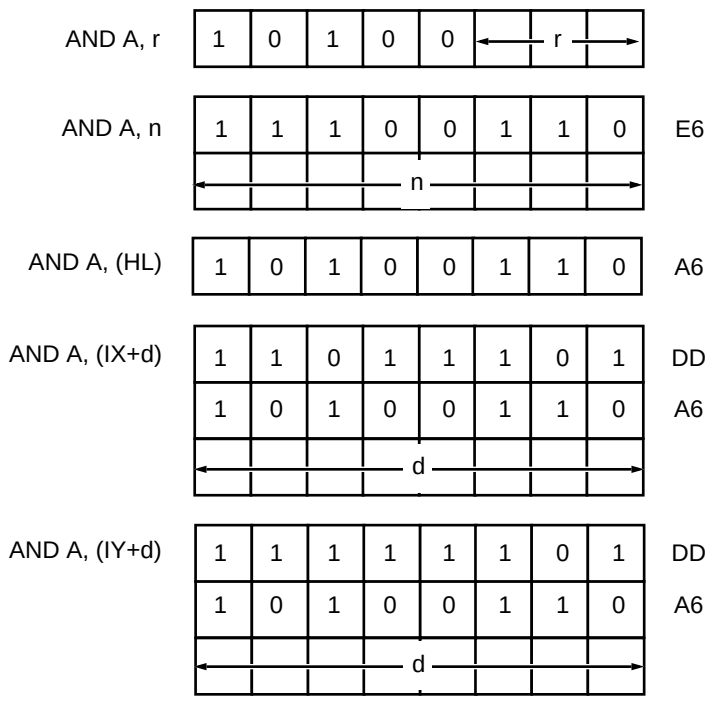
```
ADD  IX, BC
```

IX will contain 1336H.

AND A, s

Operation: A ← A and s

Format:



Description: The s operand is logically ANDed with the value in the Accumulator (A), the result is stored in A, and the flags are set as described below. "s" can be any of a register r, an "immediate" value n in the instruction itself, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects a source register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z180x States
AND A, r	1	4	4 (3 + 1 int)
AND A, n	2	7 (4, 3)	6 (3, 3)
AND A, (HL)	2	7 (4, 3)	6 (3, 3)
AND A, (IX+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)
AND A, (IX+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)

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 and the Accumulator contains C3H after the execution of

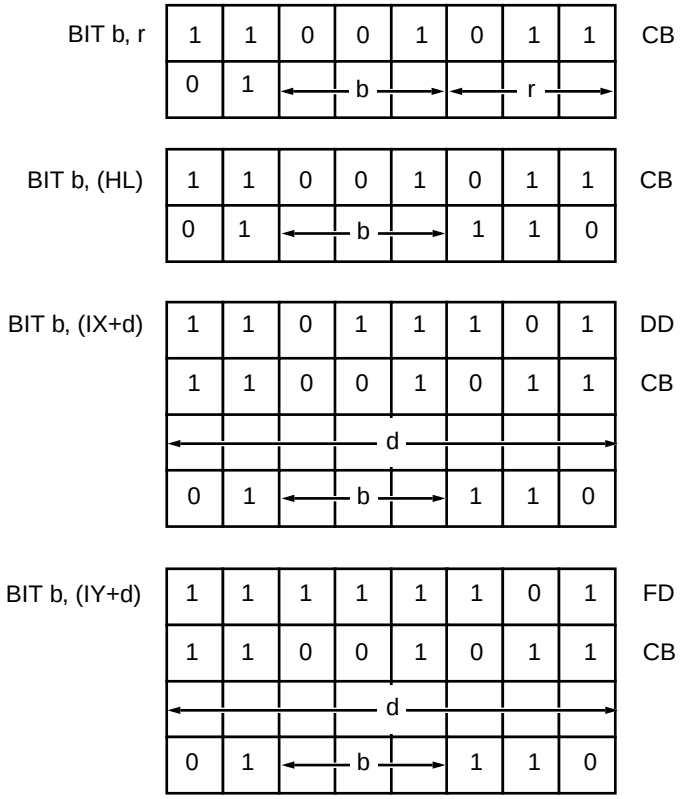
```
AND A,B
```

the Accumulator will contain 43H.

BIT b, m

Operation: Z flag ← NOT (bit b of m)

Format:



Description: The Z flag is set if bit b of operand m is zero, or cleared if the bit is 1. b can be 0 for the LS bit through 7 for the MS bit. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
BIT b, r	2	8 (4, 4)	6 (3, 3)
BIT b, (HL)	3	12 (4, 4, 4)	9 (3, 3, 3)
BIT b, (IX+d)	5	20 (4, 4, 3, 5, 4)	15 (3, 3, 3, 3, 3)
BIT b, (IY+d)	5	20 (4, 4, 3, 5, 4)	15 (3, 3, 3, 3, 3)

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 IY contains 1000H, and memory location 0FF0H contains BFH, then after the execution of

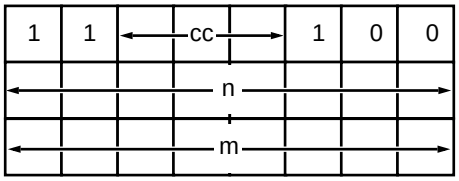
```
BIT  6, (IY - 16)
```

the Z flag will be 1. (For any of the other seven bits in IY-16, the Z flag would be 0.)

CALL cc, mn

Operation: IF cc true: (SP–1) ← PC_H, SP ← SP–2, (SP–2) ← PC_L, PC ← mn

Format:



Note: The n value in the assembled object code above is the less significant byte of the 2-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 stack, then loads the value mn 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. Because this is a 3-byte instruction, the Program Counter will have been incremented by three before the push is executed. Condition cc is programmed as one of eight values which correspond to condition bits in the Flag Register (register F). These eight status are defined in the table below:

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

Timing: If cc is true:

Z80 M Cycles	Z80 T States	Z180x M Cycles	Z180x T States
5	17 (4, 3, 4, 3, 3)	5	16 (3, 3, 3, 1 int, 3, 3)

If cc is false:

Z80 M Cycles	Z80 T States	Z180x M Cycles	Z180x T States
3	10 (4, 3, 3)	2	6 (3, 3)

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 the instruction

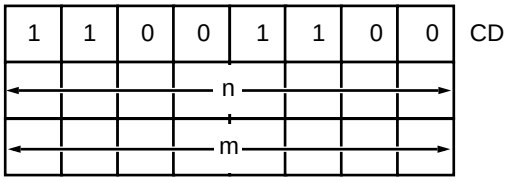
CALL NC, 2135H

is located in locations 1A47H -1A49H, then 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 mn

Operation: (SP–1) ← PC_H, (SP–2) ← PC_L, SP ← (SP)–2, PC ← mn

Format:



Note: The n value in the assembled object code above is the less significant byte of the 2-byte memory address.

Description: The current contents of the Program Counter (PC) are pushed onto the top of the stack. The value mn is then loaded into the 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 the PC.) The push is accomplished by first decrementing the current contents of the Stack Pointer, 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. Because this is a 3-byte instruction, the Program Counter will have been incremented by three before the push is executed.

Timing:

M Cycles	Z80 T States	Z180x T States
5	17 (4, 3, 4, 3, 3)	16 (3, 3, 3, 1 int, 3, 3)

Condition Bits Affected:
None.

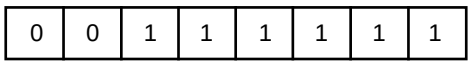
Example: If the contents of the Program Counter are 1A47H, the contents of the Stack Pointer are 3002H, and the instruction

CALL 2135H

is located at adddresses 1A47H -1A49H 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 ← /CY **Format:**



Description: The Carry flag in the F register is inverted. **Timing:**

M Cycles	Z80 T States	Z18x T States
1	4	3

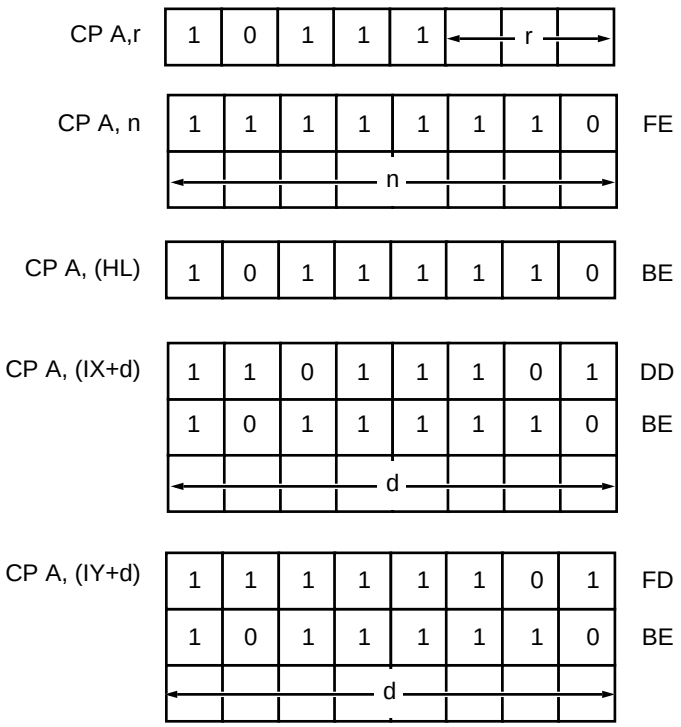
Condition Bits Affected:

S: Not affected
Z: Not affected
H: Previous carry will be copied
P/V: Not affected
N: Reset
C: Reset otherwise

CP A, s

Operation: A – s

Format:



Description: The operand s is subtracted from the contents of the Accumulator (A), the result is discarded so that A is not affected, but the result is reflected in the flags, which are set as described below. "s" can be any of a register r, an "immediate" value n in the instruction itself, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selectes a source register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
CP A, r	1	4	4 (3 + 1 int)
CP A, n	2	7 (4, 3)	6 (3, 3)
CP A, (HL)	2	7 (4, 3)	6 (3, 3)
CP A, (IX+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)
CP A,(IY+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Set if borrow from bit 4; reset otherwise
P/V: Set if overflow; reset otherwise
N: Set
C: Set if borrow; reset otherwise

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

CP A, (HL)

will result in the the following Flags:

S=0, Z=0, H=0, P/V=0, N=1, C=0

CPD

Operation: A –(HL), HL ← HL –1, BC ← BC –1

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	0	1	0	0	1	A9

Description: The contents of the memory location addressed by the HL register pair is compared with the contents of the Accumulator. In case of equality, the Z flag is set. HL and the Byte Counter (register pair BC) are decremented, and the P/V flag is set to reflect whether BC has been decremented to zero.

Timing:

M Cycles	Z80 T States	Z180x T States
3	16 (4, 4, 3, 5 int)	12 (3, 3, 3, 3 int)

Condition Bits Affected:

S: Set if A-(HL) is negative; reset otherwise
Z: Set if A = (HL); reset otherwise
H: Set if borrow from bit 4; reset otherwise
P/V: Reset if BC is now 0000; set otherwise
N: Set
C: Not Affected:

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

CPD

BC 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:

1	1	1	0	1	1	0	1	ED
1	0	1	1	1	0	0	1	B9

Description: The contents of the memory location addressed by the HL register pair is compared with the contents of the Accumulator. In case of equality, the Z flag is set. The HL and BC (Byte Counter) register pairs are decremented, and the P/V flag is set to reflect whether BC has been decremented to zero. 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 instruction is repeated. Interrupts will be recognized and refresh cycles may be executed after each data transfer. Note that if BC is set to zero prior to instruction execution, the instruction will loop through 64 Kbytes if no match is found.

Timing: For eash repetition with BC ≠ 0 and A ≠ (HL):

M Cycles	Z80 T States	Z180x T States
3	21 (4, 4, 3, 10 int)	14 (3, 3, 3, 5 int)

For BC = 0 or A = (HL):

M Cycles	Z80 T States	Z180x T States
3	16 (4, 4, 3, 5 int)	12 (3, 3, 3, 3 int)

Condition Bits Affected:

S: Set if A-(HL) is negative; reset otherwise
Z: Set if A = (HL); reset otherwise
H: Set if borrow from bit 4; reset otherwise
P/V: Reset if BC is now 0000; set otherwise
N: Set
C: Not Affected:

Example: If the HL register pair contains 1118H, the Accumulator contains F3H, BC contains 0007H, and memory locations have these contents.

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

Then after the execution of

CPDR

the contents of HL will be 1115H, the contents of BC will be 0004H, and the P/V and Z flags in the F register will both be set.

CPI

Operation: A– (HL), HL ← HL +1, BC ← BC –1

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	0	0	0	0	0	A1

Description: The contents of the memory location addressed by the HL register are compared with the contents of the Accumulator. In case of equality, the Z flag is set. Then HL is incremented, the Byte Counter (register pair BC) is decremented, and the P/V bit is set to reflect whether BC has been decremented to zero.

Timing:

M Cycles	Z80 T States	Z180x T States
3	16 (4, 4, 3, 5 int)	12 (3, 3, 3, 3 int)

Condition Bits Affected:

S: Set if A-(HL) is negative; reset otherwise
Z: Set if A = (HL); reset otherwise
H: Set if borrow from bit 4; reset otherwise
P/V: Reset if BC is now 0000; set otherwise
N: Set
C: Not affected

Example: If the HL register pair contains 1111H, memory location 1111H contains 3BH, the Accumulator contains 3BH, and BC contains 0001H. Then after the execution of

CPI

BC 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. Neither the Accumulator or memory location 1111H are affected.

CPIR

Operation: A–(HL), HL ← HL+1, BC ← BC–1

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	1	0	0	0	1	B1

Description: The contents of the memory location addressed by the HL register pair is compared with the contents of the Accumulator. In case of equality, the Z flag is set. HL is incremented and the Byte Counter (register pair BC) is decremented. The P/V flag is set to reflect whether BC has been decremented to zero. 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 instruction is repeated. Interrupts will be recognized and refresh cycles may be executed after each data transfer. Note that if BC is set to zero before instruction execution, the instruction will loop through 64 Kbytes if no match is found.

Timing: For each repetition with BC ≠ 0 and A ≠ (HL):

M Cycles	Z80 T States	Z18x T States
3	21 (4, 4, 3, 5, 10 int)	14 (3, 3, 3, 5 int)

For BC = 0 or A = (HL):

M Cycles	Z80 T States	Z18x T States
3	16 (4, 4, 3, 5 int)	12 (3, 3, 3, 3 int)

Condition Bits Affected:

S: Set if A-(HL) is negative; reset otherwise
Z: Set if A = (HL); reset otherwise
H: Set if borrow from bit 4; reset otherwise
P/V: Reset if BC now 0000; set otherwise
N: Set
C: Not Affected:

Example: If the HL register pair contains 1111H, the Accumulator contains F3H, BC 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, BC will contain 0004H, and the P/V and Z flags in the F register will both be set.

CPL

Operation: A ← /A

Format:

0	0	1	0	1	1	1	1	2F

Description: The contents of the Accumulator (register A) are inverted (one's complemented).

Timing:

M Cycles	Z80 T States	Z180x T States
1	4	3

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 D4H:

1	0	1	1	0	1	0	0

after the execution of

CPL

the Accumulator contents will be 4BH:

0	1	0	0	1	0	1	1

DAA

Operation: Decimal Adjust Accumulator

Format:

0	0	1	0	0	1	1	1	27

Description: This instruction conditionally adjusts the Accumulator after addition and subtraction of BCD values. For addition (ADD, ADC, INC) or subtraction (SUB, SBC, DEC, NEG), the following table indicates the 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	9-0	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	6-7	1	6-F	9A	1

Timing:

M Cycles	Z80 T States	Z18x T States
1	4	4 (3 + 1 int)

Condition Bits Affected:

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

Example: If 15 (BCD) and 27 (BCD) are added, 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 DAA instruction adjusts this result so that the correct BCD representation is obtained:

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

DEC ee

Operation: ee ← ee – 1

Format:

DEC ss (ss = BC, DE, HL, or SP)	0	0	s	s	1	0	1	1	
DEC IX	1	1	0	1	1	1	0	1	DD
	0	0	1	0	1	0	1	1	2B
DEC IY	1	1	1	1	1	1	0	1	FD
	0	0	1	0	1	0	1	1	2B

Description: 16-bit register ee is decremented by 1. The flags are not affected. "ee" can be any of the register pairs BC, DE, or HL, the Stack Pointer SP, or an index register IX or IY. In the first form shown above, "ss" is encoded as follows:

Register	ss
BC	00
DE	01
HL	10
SP	11

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
DEC ss	1	6 (4, 2 int)	4 (3, 1 int)
DEC IX	2	10 (4, 4, 2 int)	7 (3, 3, 1 int)
DEC IY	2	10 (4, 4, 2 int)	7 (3, 3, 1 int)

Condition Bits Affected:
None

Example: If BC contains 0 and the C flag is 0, then after the execution of

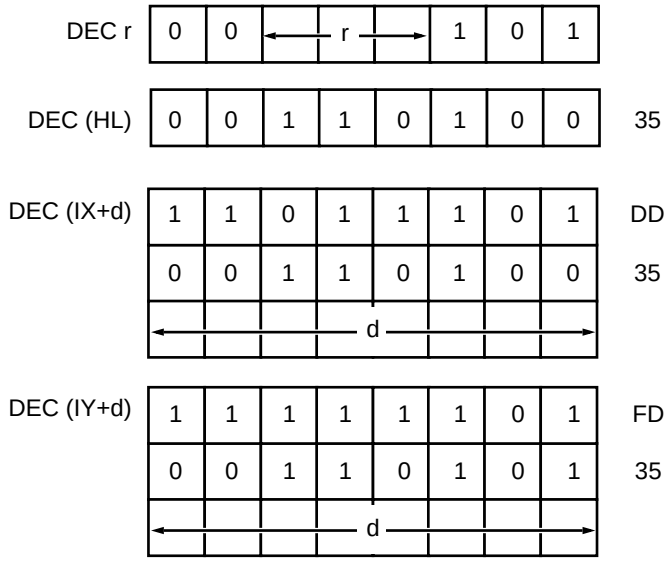
```
DEC  BC
```

BC will contain FFFFH, and the C flag will still be 0.

DEC m

Operation: m ← m – 1

Format:



Description: The 8-bit value in the m operand is decremented by one, and the flags (except C) are set as described below. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
DEC r	1	4	4 (3 + 1 int)
DEC (HL)	3	11 (4, 4, 3)	10 (3, 3, 1 int, 3)
DEC (IX+d)	5	23 (4, 4, 3, 5 int, 4, 3)	18 (3, 3, 3, 2 int, 3, 1 int, 3)
DEC (IY+d)	5	23 (4, 4, 3, 5 int, 4, 3)	18 (3, 3, 3, 2 int, 3, 1 int, 3)

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Set if borrow from bit 4, reset otherwise
P/V: Set if 80H to 7FH; reset otherwise
N: Set
C: Not affected

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

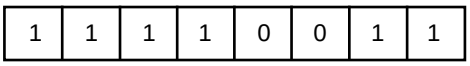
```
DEC D
```

register D will contain 29H.

DI

Operation: IFF ← 0

Format:



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

Timing:

M Cycles	Z80 T States	Z180x T States
1	4	3

Condition Bits Affected:
None.

Example: When the CPU executes the instruction

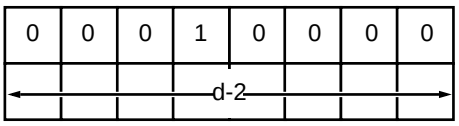
```
DI
```

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 until that time.

DJNZ, d

Operation: Decrement and Jump if Non-Zero

Format:



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 d 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 automatically adjusts for the twice incremented PC.

If decrementing leaves B with a zero value, the next instruction to be executed is taken from the location following this instruction.

Timing: if B ≠ 0:

M Cycles	Z80 T States	Z18x T States
2	13 (5 ,3, 5 int)	9 (3, 1 int, 3, 2 int)

If B = 0:

M Cycles	Z80 T States	Z18x T States
2	8 (5 ,3)	7 (3, 1 int, 3)

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      ODH         ;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:

1	1	1	1	1	0	1	1	FB

Description: The enable interrupt instruction sets both interrupt enable flip flops (IFFI and IFF2) to '1' allowing recognition of any maskable interrupt. Note that during the execution of this instruction and the following instruction, maskable interrupts not yet be enabled.

Timing:

M Cycles	Z80 T States	Z18x T States
1	4	3

Condition Bits Affected:
None.

Example: When the CPU executes instruction

```
EI
RETI
```

the maskable interrupt will be enabled after the execution of the RETI instruction.

EX AF, AF'

Operation: AF ↔ AF'

Format:

0	0	0	0	1	0	0	0	08

Description: The 2-byte contents of the register pairs AF and AF' are exchanged. (Note: register pair AF^1 consists of registers A' and F')

Timing:

M Cycles	Z80 T States	Z18x T States
1	4	4 (3, 1 int)

Condition Bits Affected:
All.

Example: If the content of register pair AF is 9900H, and the content of register pair AF^1 is 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:

1	1	1	0	1	0	1	1	EB

Description: The 2-byte contents of register pairs DE and HL are exchanged. No flags are affected.

Timing:

M Cycles	Z80 T States	Z18x T States
1	4	3

Condition Bits Affected:
None.

Example: If the content of register pair DE is 2822H, and the content of register pair HL is 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.

EX (SP), rr

Operation: $rr_L \leftrightarrow (SP)$, $rr_H \leftrightarrow (SP+1)$

Format:

EX (SP), HL	1	1	1	0	0	0	1	1	E3
EX (SP), IX	1	1	0	1	1	1	0	1	DD
	1	1	1	0	0	0	1	1	E3
EX (SP), IY	1	1	1	1	1	1	0	1	FD
	1	1	0	0	0	0	1	1	E3

Description: The contents of 16-bit register rr are exchanged with the two bytes at the top of the Stack (the address in the Stack Pointer SP and the next higher one) in memory. Neither SP nor any flags are affected. rr can be the register pair HL or the index register IX or IY.

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
EX (SP), HL	5	19 (4, 3, 4, 3, 5)	16 (3, 3, 3, 1 int, 3, 3)
EX (SP), IX	6	23 (4, 4, 3, 4, 3, 5)	19 (3, 3, 3, 3, 1 int, 3, 3)
EX (SP), IY	6	23 (4, 4, 3, 4, 3, 5)	19 (3, 3, 3, 3, 1 int, 3, 3)

Condition Bits Affected:
None

Example: If HL contains 7012H, SP contains 8856H, memory location 8856H contains 11H, and location
8857H contains 22H, then after execution of

EX (SP), HL

HL contains 2211H, memory location 8856H contains 12H, location 8857H contains 70H, and SP still contains 8856H.

EXX

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

Format:

1	1	0	1	1	0	0	1	D9

Description: Each 2-byte value in register pairs BC, DE, and HL is exchanged with the 2-byte value in BC', DE', and HL', respectively. No flags are affected.

Timing:

M Cycles	Z80 T States	Z18x T States
1	4	3

Condition Bits Affected:
None.

Example: If the contents of register pairs BC, DE, and HL are 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: Halt the processor

Format:

0	1	1	1	0	1	1	0	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.

Timing:

M Cycles	Z80 T States	Z18x T States
1	(indefinitely) 4 minimum	(indefinitely) 3 minimum

Condition Bits Affected:
None.

IM m

Operation: Set interrupt mode n (0 to 2)

Format:

1	1	1	0	1	1	0	1	ED
0	1	0	i	i	0	0	0	

Description: The processor is placed in interrupt mode n. n can be 0 to 2.

In interrupt mode 0, an interrupting device can insert any instruction onto the data bus for execution by the CPU. The first byte of the instruction is read during the interrupt acknowledge sequence. For multi-byte instructions, special external hardware is required because bytes after the first are read by normal memory read cycles. In interrupt mode 1, the processor responds to all interrupts similarly to an RST 38H instruction. In interrupt mode 2, the processor responds to interrupts by fetching an interrupt vector from the interrupting device. It then reads a byte from memory at the address having the vector as the LSbyte and the contents of the I register as the MSbyte, and loads the byte into the LSbyte of the Program Counter PC. Then it reads a second byte from memory at the next higher address, and loads that byte into the MSbyte of PC. Finally the processor starts execution of the interrupt service routine, at the address fetched from memory.

n=0 to 2 is encoded into ii in the instruction as shown below.

Mode n	ii
0	00
1	10
2	11

Timing:

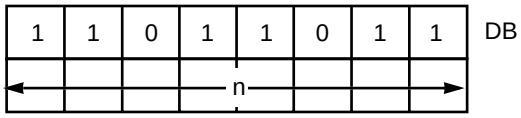
M Cycles	Z80 T States	Z18x T States
2	8 (4, 4)	6 (3, 3)

Condition Bits Affected:
None

IN A, (n)

Operation: A ← (n)

Format:



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 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. No flags are affected.

Timing:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
3	11 (4, 3, 4)	9 (3, 3, 3)	10 (3, 3, 4)

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 2301H, then after the execution of

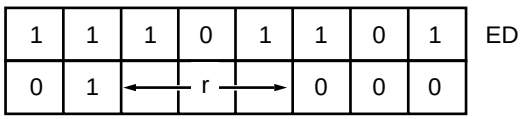
IN A, (01H)

the Accumulator will contain 7BH. If the peripheral device ignores A8 through A15, the above statement should read "...at the peripheral device mapped to I/O port address 01H..."

IN r,(C)

Operation: r ← (C)

Format:



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 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. The flags are set to reflect the input data.

Register	r
Flag	110 – Undefined opcode, set the flag
B	000
C	001
D	010
E	011
H	100
L	101
A	111

Timing:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
3	12 (4, 4, 4)	9 (3, 3, 3)	10 (3, 3, 4)

Condition Bits Affected:

S: Set if bit 7 of the input data is 1; reset otherwise
Z: Set if all 8 bits of the input data are zero; reset otherwise
H: Reset
P/V: Set if the parity of the input data 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 1007H, then after the execution of

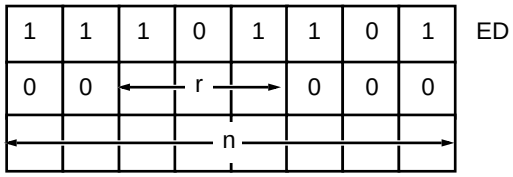
IN D, (C)

register D will contain 7BH. If the peripheral device ignores A8 through A15, the above statement should read "...at the peripheral device mapped to I/O port address 07H..."

IN0 r, (n)

Operation: r ← (0, n)

Format:



Description: The operand n is placed on A0 through A7 with zero on A8 through A15, to select an input port. The contents of the selected port are then read into the register selected by r, as shown in the following table. The flags are affected as shown below.

Register	r
B	000
C	001
D	010
E	011
H	100
L	101
(none)	110 (but the flags are still set as shown below)
A	111

Timing:

M Cycles	T States (180 reg)	T States(other 18x)
4	12 (3, 3, 3, 3)	13 (3, 3, 3, 4)

Condition Bits Affected:

S: Set if bit 7 of the input data is one, reset otherwise
Z: Set if all 8 bits of the input data are 0, reset otherwise
H: Reset
P/V: Set if the parity of the input data is even; reset otherwise
N: Reset
C: Not affected

Example: If the contents of an 8018x processor's Input/Output Control Register (IOCR) are 1FH, the 16-bit I/O address of the IOCR is 003FH, then after the execution of

```
IN0  E, (3FH)
```

the E register will contain 1FH. Note also that if the contents of the A register are non-zero, the execution of

```
IN  A, (3FH)
```

will **not** return the contents of the IOCR!

INC ee

Operation: ee ← ee + 1

Format:

INC ss (ss = BC, DE, HL, or SP)	0	0	s	s	0	0	1	1	
INC IX	1	1	0	1	1	1	0	1	DD
	0	0	1	0	0	0	1	1	23
INC IY	1	1	1	1	1	1	0	1	FD
	0	0	1	0	0	0	1	1	23

Description: The16-bit register ee is incremented by 1. The flags are not affected. "ee" can be any of the register pairs BC, DE, or HL, the Stack Pointer SP, or an index register IX or IY. In the first form shown above, "ss" is encoded as follows:

Register	ss
BC	00
DE	01
HL	10
SP	11

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
INC ss	1	6 (4, 2 int)	4 (3, 1 int)
INC IX	2	10 (4, 4, 2 int)	7 (3, 3, 1 int)
INC IY	2	10 (4, 4, 2 int)	7 (3, 3, 1 int)

Condition Bits Affected:
None

Example: If SP contains FFFFH and the C flag is 0, then after the execution of

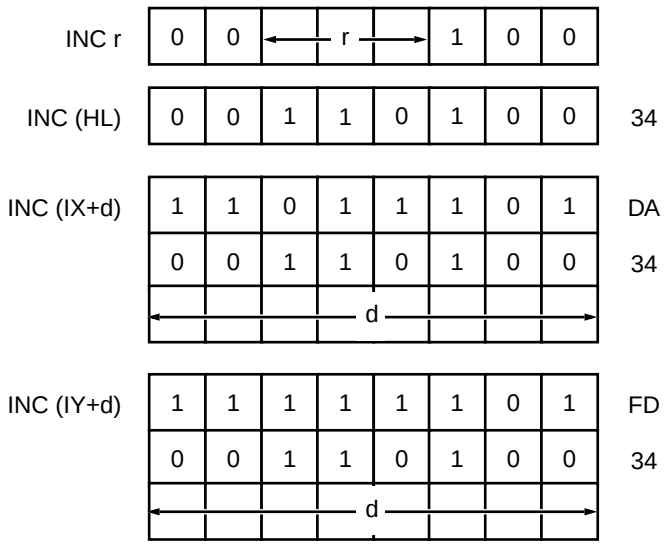
```
INC  SP
```

SP will contain 0000, and the C flag will still be 0.

INC m

Operation: m ← m + 1

Format:



Description: The 8-bit value in the m operand is incremented by one, and the flags (except C) are set as described below. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
INC r	1	4	4 (3, 1 int)
INC (HL)	3	11 (4, 4, 3)	10 (3, 3, 1 int, 3)
INC (IX+d)	5	23 (4, 4, 3, 5 int, 4, 3)	18 (3, 3, 3, 2 int, 3, 1 int, 3)
INC (IY+d)	5	23 (4, 4, 3, 5 int, 4, 3)	18 (3, 3, 3, 2 int, 3, 1 int, 3)

Condition Bits Affected:

S: Set if result is negative, reset otherwise
Z: Set if result is 0; reset otherwise
H: Set if carry from bit 3; reset otherwise
P/V: Set if 7FH to 80H; reset otherwise
N: Reset
C: Not affected

Example: If IY contains 2020H, and memory location 2030H contains 34H, after the execution of INC (IY + 10H) memory location 2030H contains 35H.

IND

Operation: (HL) ← (C), B ← B – 1, HL ← HL – 1

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	0	1	0	1	0	AA

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 is read from the selected port and is captured by the CPU. The contents of the HL register pair are placed on the address bus and the input byte is written to the corresponding location of memory. Finally the byte counter and register pair HL are decremented.

Note: Note that this instruction is not particularly compatible with peripheral devices that decode A15-8 as part of a 16-bit I/O address.

Timing:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	16 (4, 5, 3, 4)	12 (3, 3, 3, 3)	13 (3, 3, 4, 3)

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:

1	1	1	0	1	1	0	1	ED
1	0	1	1	1	0	1	0	BA

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 is read from the selected port and is captured by the CPU. The contents of the HL register pair are placed on the address bus and the input byte is written to 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 instruction is repeated. Interrupts will be recognized and refresh cycles may be executed after each data transfer. Note that if B is set to zero prior to instruction execution, 256 bytes of data will be input.

Note: This instruction is not particularly compatible with peripheral deivces that decode A15-8 as part of a 16-bit I/O address.

Timing: If B ≠ 0

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	21 (4, 5, 3, 4, 5 int)	14 (3, 3, 3, 3, 2 int)	15 (3, 3, 4, 3, 2 int)

If B = 0:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	16 (4, 5, 3, 4)	12 (3, 3, 3, 3)	13 (3, 3, 4, 3)

Condition Bits Affected:

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

Example: If register C contains 07H, register B contains 03H, the HL register pair contains1000H, and the following sequence of bytes are available at the peripheral device mapped to I/O port address 07H:

51H (first)
A9H
03H (third)

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

INI

Operation: (HL) ← (C), B ← B – 1, HL ← HL + 1

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	0	0	0	1	0	A2

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 is read from the selected port and is captured by the CPU. The contents of the HL register pair are then placed on the address bus and the input byte is written to the corresponding location of memory. Finally the byte counter is decremented and register pair HL is incremented.

Note: This instruction is not particularly compatible with peripheral devices that decode A15-8 as part of a 16-bit I/O address.

Timing:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	16 (4, 5, 3, 4)	12 (3, 3, 3, 3)	13 (3, 3, 4, 3)

Condition Bits Affected:

S: Unknown
Z: Set if B now 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:

1	1	1	0	1	1	0	1	ED
1	0	1	1	0	0	1	0	B2

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 is read from the selected port and is captured by the CPU. The contents of the HL register pair are placed on the address bus and the input byte is written to the corresponding location of memory. Then register pair HL is incremented and the byte counter is decremented. If decrementing causes B to go to zero, the instruction is terminated. If B is not zero, and the instruction is repeated. Interrupts will be recognized and refresh cycles be executed after each data transfer. Note that if B is set to zero prior to instruction execution, 256 bytes of data will be input.

Note: This instruction is not particularly compatible with peripheral devices that decode A15-8 as part of a 16-bit I/O address.

Timing: If B ≠ 0:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	21 (4, 5, 3, 4, 5 int)	14 (3, 3, 3, 3, 2 int)	15 (3, 3, 4, 3, 2 int)

If B = 0:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	16 (4, 5, 3, 4)	12 (3, 3, 3, 3)	13 (3, 3, 4, 3)

Condition Bits Affected:

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

Example: If register C contains 07H, register B contains 03H, the HL register pair contains1000H, and the following sequence of bytes are available at the peripheral device mapped to I/O port address 07H:

51H (first)
A9H
03H (third)

then 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 (rr)

Operation: PC ← rr

Format:

JP (HL)	1	1	1	0	1	0	0	1	E9
JP (IX)	1	1	0	1	1	1	0	1	DD
	1	1	1	0	1	0	0	1	E9
JP (IY)	1	1	1	1	1	1	0	1	FD
	1	1	1	0	1	0	0	1	E9

Description: The contents of 16-bit register rr are loaded into the Program Counter PC, so that the next instruction starts at that address. No flags are affected. rr can be the register pair HL or the index register IX or IY.

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
JP (HL)	1	4	3
JP (IX)	2	8 (4, 4)	6 (3, 3)
JP (IY)	2	8 (4, 4)	6 (3, 3)

Condition Bits Affected:
None

Example: If HL contains 7404H, then after execution of

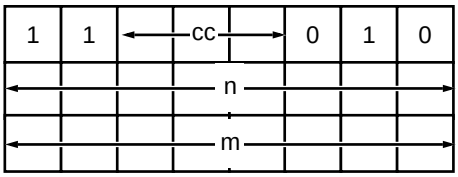
```
JP (HL)
```

the processor starts fetching the next instruction from memory location 7404H.

JP cc, mn

Operation: IF cc is true, PC← mn
else continue

Format:



Note: The n value above is the low order byte of the 2-byte memory address.

Description: If condition cc is true, the instruction loads the value mn into the Program Counter, and the program continues with the instruction beginning at address mn. 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 which correspond to condition bits in the Flag Register (register F):

cc	Condition	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

Timing: If jump is taken:

Z80 M Cycles	Z80 T States	Z18x M Cycles	Z18x T States
3	10 (4, 3, 3)	3	9 (3, 3, 3)

If jump is not taken:

Z80 M Cycles	Z80 T States	Z18x M Cycles	Z18x T States
3	10 (4, 3, 3)	2	6 (3, 3)

Condition Bits Affected:
None.

Example: If the Carry flag (C flag in the F register) is set after the execution of

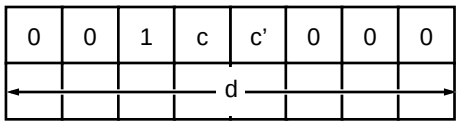
JP C, 1520H

the Program Counter will contain 1520H.

JR cc', d

Operation: If cc' true, PC ← PC +/-d
else continue

Format:



Description: If the flag condition selected by cc' is true, the 8-bit signed displacement d is added to the Program Counter PC, which by this time has been incremented to the address of the next instruction. If the selected condition is false, execution continues with the next instruction. No flags are affected. cc' is encoded to express the following flag conditions:

cc'	Condition
00	NZ (nonzero)
01	Z (zero)
10	NC (no carry)
11	C (carry)

Timing: If the condition is met and a jump occurs:

M Cycles	Z80 T States	Z18x T States
2	12 (4, 3, 5 int)	8 (3, 3, 2 int)

If the condition is not met, and no jump occurs:

M Cycles	Z80 T States	Z18x T States
2	7 (4, 3)	6 (3, 3)

Condition Bits Affected:
None

Example: Because the assembler symbol $ refers to the first byte of the current instruction, the assembler will assemble the instruction

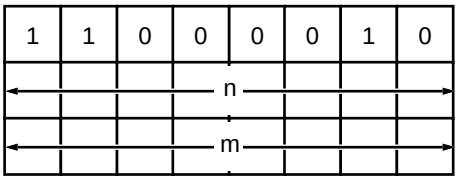
```
JR  C, $ - 4
```

as 38H FAH. (The assembled d byte is always 2 less than a displacement from $ in the source; FAH = -6 is two less than the -4 given in the source operand.) If C is set and the preceding instruction has its opcode byte located at location 480H, execution of the instruction will result in a branch to 47CH.

JP mn

Operation: PC ← mn

Format:



Note: The n value above is the low order byte of the two-byte address.

Description: Operand mn is loaded into register pair PC (Program Counter). The next instruction is fetched from the location designated by the new contents of the PC.

Timing:

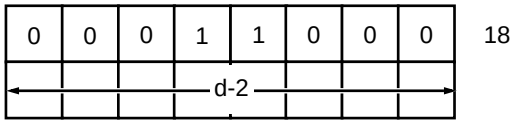
M Cycles	Z80 T States	Z18x T States
3	10 (4, 3, 3)	9 (3, 3, 3)

Condition Bits Affected:
None.

JR d

Operation: PC ← PC +/-d

Format:



Description: This instruction provides for unconditional branching to other segments of a program. The value of the displacement d 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 automatically adjusts for the twice incremented PC.

Timing:

M Cycles	Z80 T States	Z18x T States
2	12 (4, 3, 5 int)	8 (3, 3, 2 int)

Condition Bits Affected:
None.

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

JR $+5

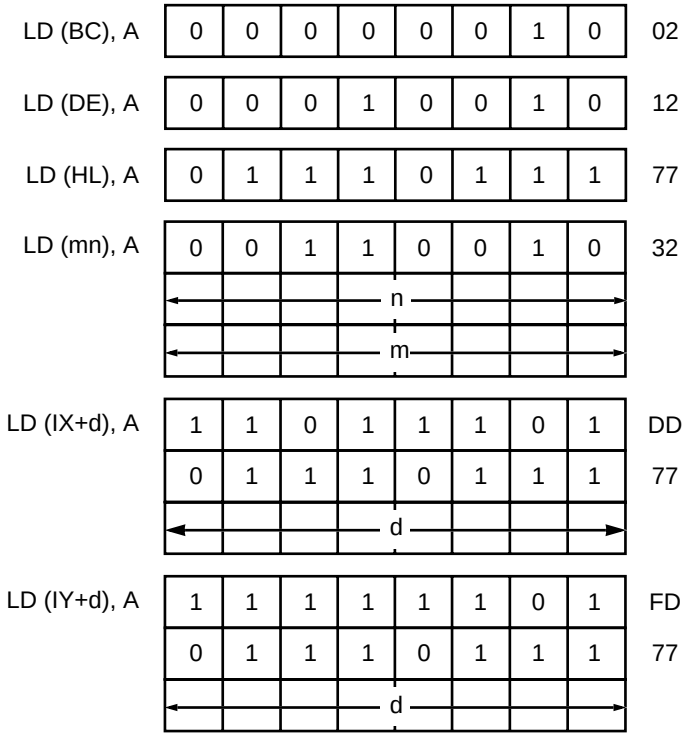
The resulting object code and final PC value is shown below:

Location	Instruction
480	18
481	03
482	–
483	–
484	–
485	← PC after jump

LD (aa), A

Operation: (aa) ← A

Format:



Description: The value in the Accumulator A is loaded into the memory location selected by aa. The flags are not affected. The memory address "aa" can be the contents of any of the register pairs BC, DE, or HL, a direct address mn in the instruction itself, or the sum of the contents of an index register IX or IY and a signed 8-bit displacement d.

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
LD (BC), A	2	7 (4, 3)	7 (3, 1 int, 3)
LD (DE), A	2	7 (4, 3)	7 (3, 1 int, 3)
LD (HL), A	2	7 (4, 3)	7 (3, 1 int, 3)
LD (mn), A	4	13 (4, 3, 3, 3)	13 (3, 3, 3, 1 int, 3)
LD (IX+d), A	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)
LD (IY+d), A	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)

Condition Bits Affected:
None

Example: If the Accumulator contains D7H, the instruction

```
LD (3141H), A
```

results in D7H being written into memory location 3141H.

LD (mn), ee

Operation: (mn) ← ee_L, (mn+1) ← ee_H

Format:

LD (mn), HL

0	0	1	0	0	0	1	0	22
←				n			→	
←				m			→	

LD (mn), ss
(ss = BC, DE, or SP)

0	0	1	1	1	0	1	0	ED
0	1	s	s	0	0	1	1	
←				n			→	
←				m			→	

LD (mn), IX

1	1	0	1	1	1	0	1	DD
0	0	1	0	0	0	1	0	22
←				n			→	
←				m			→	

LD (mn), IY

1	1	1	1	1	1	0	1	FD
0	0	1	0	0	0	1	0	22
←				n			→	
←				m			→	

Description: The contents of the less-significant half of 16-bit register ee are stored in memory location mn, and the contents of the more-significant half are stored in location mn+1. The flags are not affected. "ee" can be any of the register pairs BC, DE, or HL, the Stack Pointer SP, or an index register IX or IY. "n" is the least-significant byte of the memory address. In the second form shown above, "ss" is encoded as follows:

Register	ss
BC	00
DE	01
HL	10 (but the first form shown above is preferred)
SP	11

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
LD (mn), HL	5	16 (4, 3, 3, 3, 3)	15 (3, 3, 3, 3, 3)
LD (mn), ss	6	20 (4, 4, 3, 3, 3, 3)	19 (3, 3, 3, 3, 1 int, 3, 3)
LD (mn), IX	6	20 (4, 4, 3, 3, 3, 3)	19 (3, 3, 3, 3, 1 int, 3, 3)
LD (mn), IY	6	20 (4, 4, 3, 3, 3, 3)	19 (3, 3, 3, 3, 1 int, 3, 3)

Condition Bits Affected:
None

Example: If SP contains CDE0H, then after the execution of

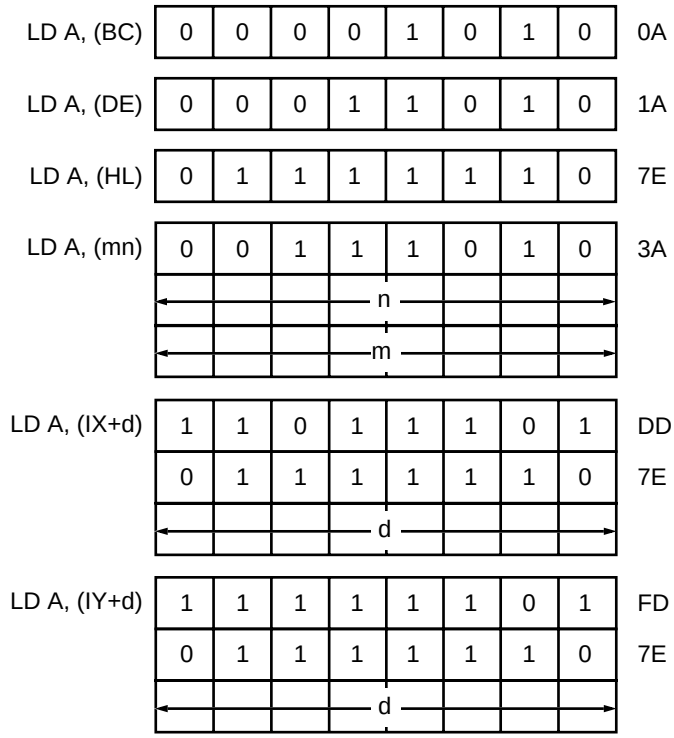
LD (6789H), SP

memory location 6789H contains E0H and location 678AH contains CDH.

LD A, (aa)

Operation: A ← (aa)

Format:



Description: The value in the memory location selected by aa is loaded into the Accumulator A. The flags are not affected. The memory address "aa" can be the contents of any of the register pairs BC, DE, or HL, a direct address mn in the instruction itself, or the sum of the contents of an index register IX or IY and a signed 8-bit displacement d.

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
LD A, (BC)	2	7 (4, 3)	6 (3, 3)
LD A, (DE)	2	7 (4, 3)	6 (3, 3)
LD A, (HL)	2	7 (4, 3)	6 (3, 3)
LD A, (mn)	4	13 (4, 3, 3, 3)	12 (3, 3, 3, 3)
LD A, (IX+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)
LD A, (IY+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)

Condition Bits Affected:
None

Example: If index register IY contains 25AFH, the instruction

LD A, (IY + 19H)

makes the processor calculate the address 25AFH + 19H, which is 25C8H. If memory location 25C8H contains 39H, the result of the instruction is that 39H is loaded into the Accumulator A.

LD A, I

Operation: A ← I

Format:

1	1	1	0	1	1	0	1	ED
0	1	0	1	0	1	1	1	57

Description: The contents of the Interrupt Vector Register I are loaded into the Accumulator, and the P/V flag is set to the state of the interrupt enable flag IEF2.

Timing:

M Cycles	Z80 T States	Z18x T States
2	9 (4, 5)	6 (3, 3)

Condition Bits Affected:

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

Note: If an interrupt occurs during execution of this instruction, the Parity flag will contain a 0.

LD A, R

Operation: A, ← R

Format:

1	1	1	0	1	1	0	1	ED
0	1	0	1	1	1	1	1	5F

Description: The contents of Memory Refresh Register R are loaded into the Accumulator, and the P/V flag is set to the state of the interrupt enable flag IEF2.

Timing:

M Cycles	Z80 T States	Z18x T States
2	9 (4, 5)	6 (3, 3)

Condition Bits Affected:

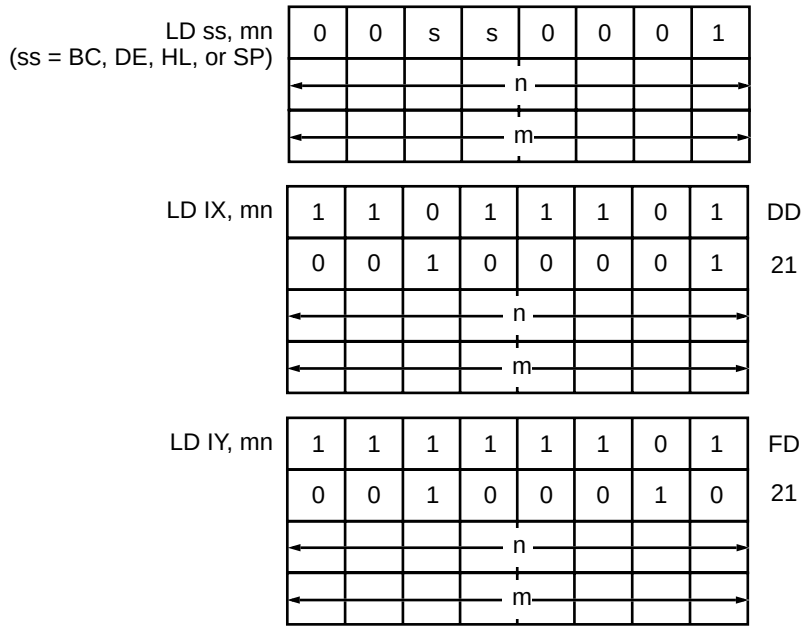
S: Set if R-Register is negative; reset otherwise
Z: Set if R-Register is zero; reset otherwise
H Reset
P/V: Contains contents of IFF2
N: Reset
C: Not affected

Note: If an interrupt occurs during execution of this instruction, the parity flag will contain a 0.

LD ee, mn

Operation: ee ← mn

Format:



Description: The 16-bit "immediate" value mn in the instruction is loaded into register ee. The flags are not affected. "ee" can be any of the register pairs BC, DE, or HL, the Stack Pointer SP, or an index register IX or IY. "n" is loaded into the less-significant half of the register. In the first form shown above, "ss" is encoded as follows:

Register	ss
BC	00
DE	01
HL	10
SP	11

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
LD ss, mn	3	10 (4, 3, 3)	9 (3, 3, 3)
LD IX, mn	4	14 (4, 4, 3, 3)	12 (3, 3, 3, 3)
LD IY, mn	4	14 (4, 4, 3, 3)	12 (3, 3, 3, 3)

Condition Bits Affected:
None

Example: After the execution of

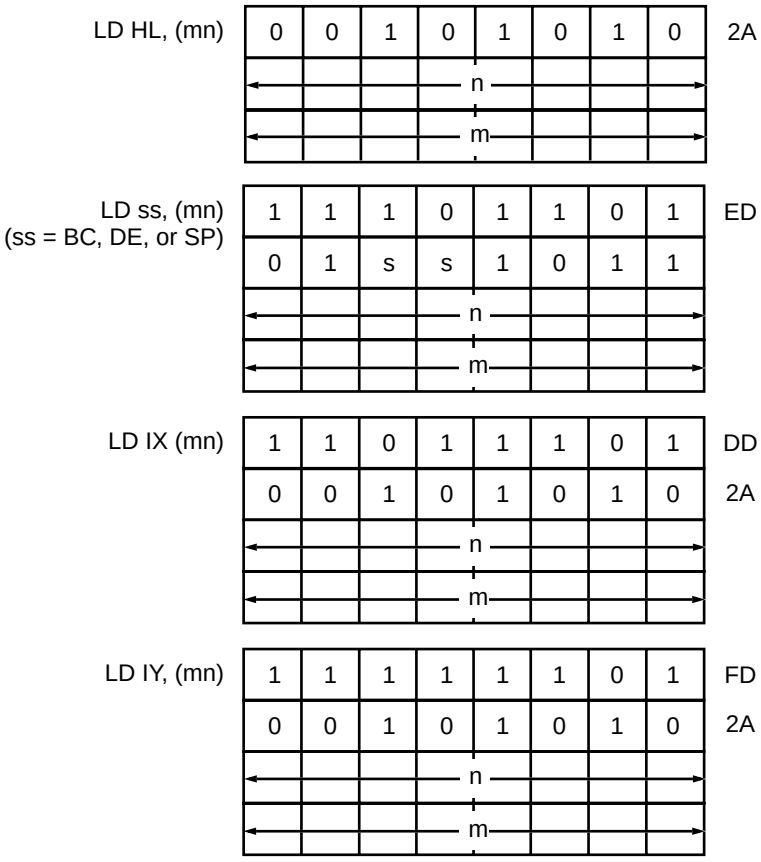
```
LD  SP, 7FFEH
```

the Stack Pointer contains 7FFEH.

LD ee, (mn)

Operation: $ee_L \leftarrow (mn)$, $ee_H \leftarrow (mn+1)$

Format:



Description: The contents of memory location mn are loaded into the less-significant half of 16-bit register ee, and the contents of location mn+1 are loaded into the more-significant half. The flags are not affected. "ee" can be any of the register pairs BC, DE, or HL, the Stack Pointer SP, or an index register IX or IY. The "n" following the op code(s) is the less-significant byte of the memory address. In the second form shown above, "ss" is encoded as follows:

Register	ss
BC	00
DE	01
HL	10 (but the first form shown above is preferred)
SP	11

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
LD HL, (mn)	5	16 (4, 3, 3, 3, 3)	15 (3, 3, 3, 3, 3)
LD ss, (mn)	6	20 (4, 4, 3, 3, 3, 3)	18 (3, 3, 3, 3, 3, 3)
LD IX, (mn)	6	20 (4, 4, 3, 3, 3, 3)	18 (3, 3, 3, 3, 3, 3)
LD IY, (mn)	6	20 (4, 4, 3, 3, 3, 3)	18 (3, 3, 3, 3, 3, 3)

Condition Bits Affected:
None

Example: If memory location 6789H contains 34H and location 678AH contains 12H, then after the execution of

```
LD  BC, (6789H)
```

B contains 12H and C contains 34H.

LD I, A

Operation: I ← A

Format:

1	1	1	0	1	1	0	1	ED
0	1	0	0	0	1	1	1	47

Description: The contents of the Accumulator are loaded into the Interrupt Control Vector Register, I. No flags are affected.

Timing:

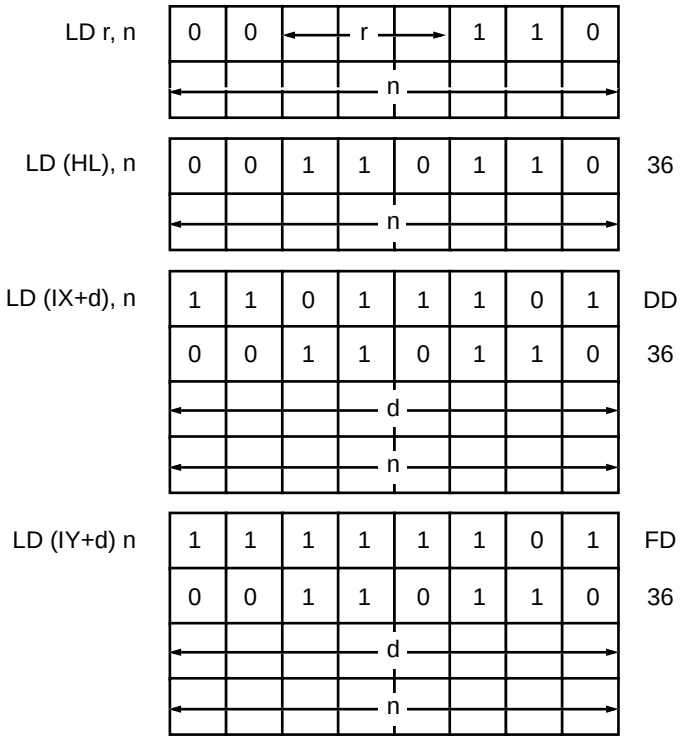
M Cycles	Z80 T States	Z18x T States
2	9 (4, 5)	6 (3, 3)

Condition Bits Affected:
None

LD m, n

Operation: m ← n

Format:



Description: The "immediate" value n is loaded into the m operand. The flags are not affected. "m" can be any of a destination register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects a destination register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
LD r, n	2	7 (4, 3)	3 (3, 3)
LD (HL), n	3	10 (4, 3, 3)	9 (3, 3, 3)
LD (IX+d), n	5	19 (4, 4, 3, 3, 2 int, 3)	15 (3, 3, 3, 3, 3)
LD (IY+d), n	5	19 (4, 4, 3, 3, 2 int, 3)	15 (3, 3, 3, 3, 3)

Condition Bits Affected:
None

Example: If the HL register pair contains 4444H, the instruction

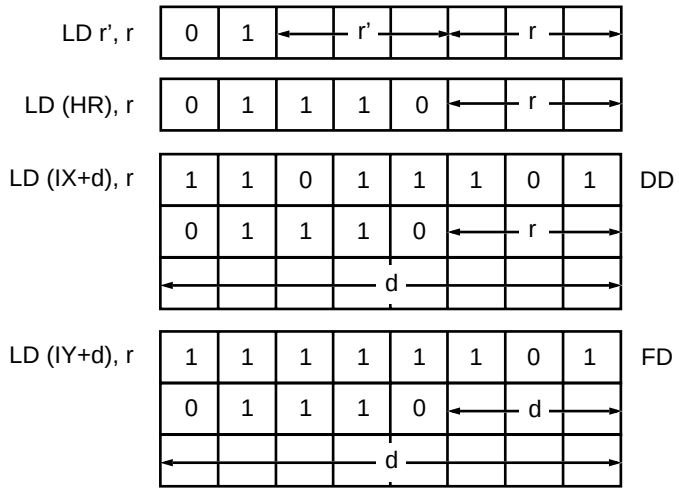
LD (HL), 28H

loads 28H into memory location 4444H

LD m, r

Operation: m ← r

Format:



Description: The value of the source register r is loaded into the m operand. The flags are not affected. "m" can be any of a destination register r' a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. r selects a source register, and in the register-to-register form r' selects a destination register, as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
LD r', r	1	4	4 (3, 1 int)
LD (HL), r	2	7 (4, 3)	7 (3, 1 int, 3)
LD (IX+d), r	4	19 (4, 4, 3, 5 int, 3)	4 (3, 3, 3, 2 int, 3)
LD (IY+d), r	4	19 (4, 4, 3, 5 int, 3)	4 (3, 3, 3, 2 int, 3)

Condition Bits Affected:
None

Example: If the C register contains 1CH, and index register IX contains 3100H, after the instruction

```
LD  (IY + 6), C
```

memory location 3106H contains 1CH.

LD R, A

Operation: R/A

Format:

1	1	1	0	1	1	0	1	ED
0	1	0	0	1	1	1	1	4F

Description: The contents of the Accumulator are loaded into the Memory Refresh register R. No flags are affected.

Timing:

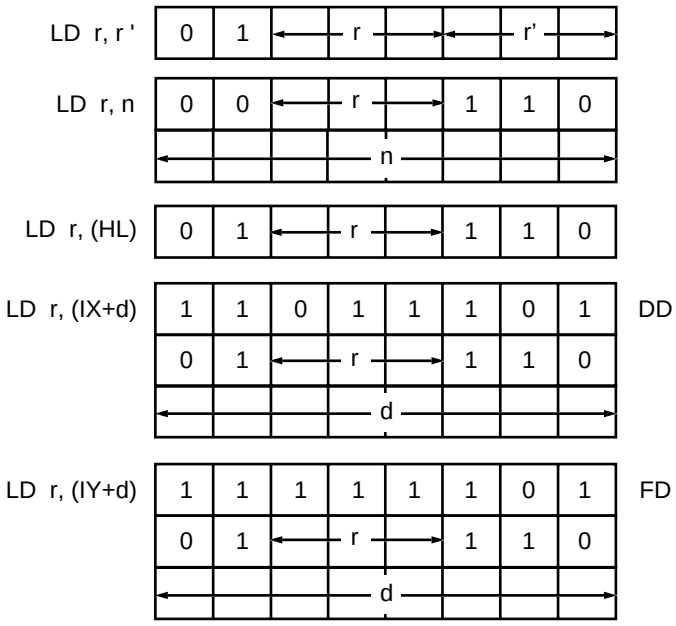
M Cycles	Z80 T States	Z18x T States
2	9 (4, 5)	6 (3, 3)

Condition Bits Affected:
None.

LD r, s

Operation: r ← s

Format:



Description: The value of the s operand is loaded into the destination register r. The flags are not affected. "s" can be any of a source register r', an "immediate" value n in the instruction itself, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. r selects a destination register, and in the register-to-register form r' selects a source register, as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
LD r, r'	1	4	4 (3, 1 int)
LD r, n	2	7 (4, 3)	6 (3, 3)
LD r, (HL)	2	7 (4, 3)	6 (3, 3)
LD r, (IX+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)
LD r, (IY+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)

Condition Bits Affected:
None

Example: If index register IY contains 25AFH, the instruction

LD B, (IY + 19H)

makes the processor calculate the address 25AFH + 19H, which is 25C8H. If memory location 25C8H contains 39H, the result of the instruction is that 39H is loaded into the B register.

LD SP, rr

Operation: SP ← rr

Format:

LD SP, HL	1	1	1	1	1	0	0	1	F9
LD SP, IX	1	1	0	1	1	1	0	1	DD
	1	1	1	1	1	0	0	1	F9
LD SP, IY	1	1	1	1	1	1	0	1	FD
	1	1	1	1	1	0	0	1	F9

Description: The contents of the 16-bit register rr are loaded into the Stack Pointer SP. The flags are not affected. "rr" can be the HL register pair or an index register IX or IY.

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
LD SP, HL	1	6 (4, 2 int)	4 (3, 1 int)
LD SP, IX	2	10 (4, 4, 2 int)	7 (3, 3, 1 int)
LD SP, IY	2	10 (4, 4, 2 int)	7 (3, 3, 1 int)

Condition Bits Affected:
None

Example: If IX contains AB04H, then after the execution of

```
LD   SP, IX
```

SP will also contains AB04H.

LDD

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

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	0	1	0	0	0	A8

Description: This 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 are decremented, as is the BC (Byte Counter) register pair. The P/V flag is set to reflect whether BC has been decremented to zero.

Timing:

M Cycles	Z80 T States	Z18x T States
4	16 (4, 4, 3, 5)	12 (3, 3, 3, 3)

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Reset if BC now 0000; set 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
repeat until (BC) is zero

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	1	1	0	0	0	B8

Description: This 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 registers, as well as the BC (Byte Counter), are decremented. If decrementing causes BC to go to zero, the instruction is terminated. If BC is not zero, the program counter is decremented by two and the instruction is repeated. Interrupts will be recognized and two refresh cycles will be executed after each data transfer. Note that if BC is set to zero prior to instruction execution, the instruction will loop through 64 Kbytes.

Timing: For each repetition while BC ≠ 0:

M Cycles	Z80 T States	Z18x T States
4	21 (4, 4, 3, 5, 5 int)	14 (3, 3, 3, 3, 2 int)

For BC = 0:

M Cycles	Z80 T States	Z18x T States
4	16 (4, 4, 3, 5)	12 (3, 3, 3, 3)

Condition Bits Affected:

S: Not Affected:
Z: Not Affected:
H: Reset
P/V: Reset
N: Reset

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:

1	1	1	0	1	1	0	1	ED
1	0	1	0	0	0	0	0	A0

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, the BC (Byte Counter) register pair is decremented, and the P/V flag reflects whether BC has been decremented to zero.

M Cycles	Z80 T States	Z18x T States
4	16 (4, 4, 3, 5)	12 (3, 3, 3, 3)

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Reset if BC now 0000; set 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

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
repeat until (BC) is zero

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	1	0	0	0	0	B0

Description: This 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 two and the instruction is repeated. Interrupts will be recognized and two refresh cycles will be executed after each data transfer. Note that if BC is set to zero prior to instruction execution, the instruction will loop through 64 Kbytes.

Timing: For each repetition while BC ≠ 0:

M Cycles	Z80 T States	Z18x T States
4	21 (4, 4, 3, 5, 5 int)	14 (3, 3, 3, 3, 2 int)

For BC = 0:

M Cycles	Z80 T States	Z18x T States
4	16 (4, 4, 3, 5)	12 (3, 3, 3, 3)

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

MLT ss

Operation: $ss \leftarrow ss_L * ss_H$

Format:

1	1	1	0	1	1	0	1	ED
0	1	s	s	1	1	0	0	

Description: The two 8-bit halves of a register pair are multiplied and the product is stored in the same register pair. Multiplication is unsigned, and the flags are not affected. "ss" is encoded as follows:

Register	ss
BC	00
DE	01
HL	10
SP	11

Timing:

M Cycles	Z80 T States	Z18x T States
2	NA	17 (3, 3, 11 int)

Condition Bits Affected:
None

Example: If H contains 08H and L contains AAH, then after the execution of

```
MLT  HL
```

HL will contain 0550H. (In decimal, this is 8 * 170 = 1360.)

NEG

Operation: A ← 0 – A

Format:

1	1	1	0	1	1	0	1	ED
0	1	0	0	0	1	0	0	44

Description: The 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.

Timing:

M Cycles	Z80 T States	Z18x T States
2	8 (4, 4)	6 (3, 3)

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Set if borrow from bit 4; reset otherwise
P/V: Set if Accumulator was 80H before operation; reset otherwise
N: Set
C: Reset if Accumulator was 00H before operation; set otherwise

Example: If the contents of the Accumulator are 98H, -104_{10}

1	0	0	1	1	0	0	0

after the execution of

NEG

the Accumulator contents will be 68H,

0	1	1	0	1	0	0	0

104_{10}

NOP

Operation: None

Format:

0	0	0	0	0	0	0	0	00

Description: The CPU performs no operation during this machine cycle.

Timing:

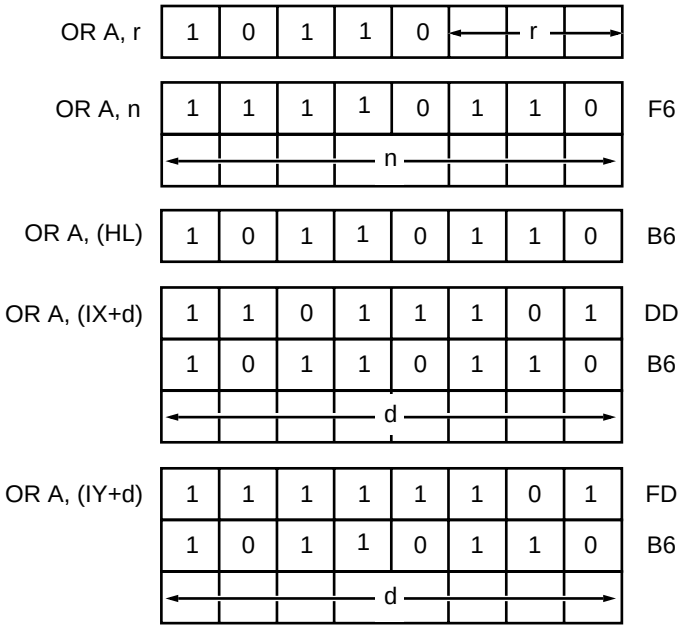
M Cycles	Z80 T States	Z18x T States
1	4	3

Condition Bits Affected:
None.

OR A, s

Operation: A ← A OR s

Format:



Description: The s operand is logically inclusive ORed with the value in the Accumulator A, the result is stored in A, and the flags are set as described below. "s" can be any of a register r, an "immediate" value n in the instruction itself, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects a source register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
OR A, r	1	4	4 (3 + 1 int)
OR A, n	2	7 (4, 3)	6 (3, 3)
OR A, (HL)	2	7 (4, 3)	6 (3, 3)
OR A, (IX+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)
OR A, (IY+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)

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: Reset

Example: If the H register contains 48H and the Accumulator contains 12H, after the execution of

OR H

the Accumulator will contain 5AH.

OTDM

Operation: (0, (C)) ← (HL)
HL ← HL - 1
C ← C - 1
B ← B - 1

Format:

1	1	1	0	1	1	0	1	ED
1	0	0	0	1	0	1	1	8B

Description: A byte is read from memory at the address in HL, and is then written to the output port selected by placing the contents of C on A7-A0 and zeroes on A15-A8 (if any). Then the memory address in HL and the I/O address in C are both decremented, a byte count in B is decremented, and the Z flag is set to reflect whether B has been decremented to zero. Other flags are affected as shown below.

Timing:

M Cycles	T States (180 reg)	T States (other 18x)
4	14 (3, 3, 3, 3, 2 int)	15 (3, 3, 3, 4, 2 int)

Condition Bits Affected:

S: Set if new (B) negative; reset otherwise
Z: Set if new (B) 0; reset otherwise
H: Set if borrow from bit 4 of B; reset otherwise
P/V: Set if parity of new (B) is even; cleared otherwise
N: Set if bit 7 of data byte was 1; reset otherwise
C: Set if B 0 to FF; reset otherwise

Example: If HL contains 4200H, C contains 80H, B contains 10H, and memory location 4200 contains 81H, after execution of the instruction

OTDM

81H will have been written to output port 0080H, HL will contain 41FFH, C will contain 7FH, B will contain 0FH, and the flags will be S=Z=C=0, H=P/V=N=1.

OTDMR

Operation: (0, (C)) ← (HL)
HL ← HL – 1
C ← C – 1
B ← B – 1
Repeat above steps until B=0

Format:

1	1	1	0	1	1	0	1	ED
1	0	0	1	1	0	1	1	9B

Description: A byte is read from memory at the address in HL, and is then written to the output port selected by placing the contents of C on A7-A0 and zeroes on A15-A8 (if any). Then the memory address in HL and the I/O address in C are both decremented, and a byte count in B is decremented. These steps are repeated until B has been decremented to 0. An initial B value of zero will cause 256 bytes to be transferred from memory to output ports. The flags are affected as shown below.

Interrupts can occur after any cycle, and the instruction will continue transparently upon return from the interrupt service routine. This instruction in useful for initializing a block of output registers, such as the 80180 "core" registers.

Timing:

	M Cycles	T States (180 reg)	T States (other 18x)
(new (B) nonzero)	4	16 (3, 3, 3, 3, 4 int)	17 (3, 3, 3, 4, 4 int)
(new (B) zero)	4	14 (3, 3, 3, 3, 2 int)	15 (3, 3, 3, 4, 2 int)

Condition Bits Affected:

S: Reset
Z: Set
H: Reset
P/V: Set
N: Set if bit 7 of last data byte was 1; reset otherwise
C: Reset

Example: If HL contains 4200H, C contains 80H, B contains 3, and memory locations 41FEH to 4200H contains 13H 00H 81H, after execution of the instruction

OTDMR

81H will have been written to output port 0080H, 00 to port 007FH, 13H to port 007EH, HL will contain 41FDH, C will contain 7DH, and B will contain 0.

OTDR

Operation: (C) ← (HL), B ← B – 1, HL ← HL – 1

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	1	1	0	1	1	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 (A7 through A0) 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 (A15 through A8) 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 instruction is repeated. If B has gone to zero, the instruction is terminated. Interrupts will be recognized and refresh cycles may be executed after each data transfer. Note that if B is set to zero prior to instruction execution, the instruction will output 256 bytes of data.

Note: This instruction is not particularly compatible with peripheral devices that decode A15-8 as part of a 16-bit I/O address.

Timing: If B ≠ 0:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	21 (4, 5, 3, 4, 5 int)	14 (3, 3, 3, 3, 2 int)	15 (3, 3, 3, 4, 2 int)

If B = 0:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	16 (4, 5, 3, 4)	12 (3, 3, 3, 3)	13 (3, 3, 3, 4)

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 memory locations have the following contents:

Register	r
0FFEH	51H
0FFFH	A9H
1000H	03H

then after the execution of

OTDR

the HL register pair will contain 0FFDH, 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 (first)
A9H
51H (third)

OTIM

Operation: (0, (C)) ← (HL)
HL ← HL + 1
C ← C + 1
B ← B –1

Format:

1	1	1	0	1	1	0	1	ED
1	0	0	0	1	0	1	1	83

Description: A byte is read from memory at the address in HL, and is then written to the output port selected by placing the contents of C on A7-A0 and zeroes on A15-A8 (if any). Then the memory address in HL and the I/O address in C are both incremented, a byte count in B is decremented, and the Z flag is set to reflect whether B has been decremented to zero. Other flags are affected as shown below.

Timing:

M Cycles	T States (180 reg)	T States (other 18x)
4	14 (3, 3, 3, 3, 2 int)	15 (3, 3, 3, 4, 2 int)

Condition Bits Affected:

S: Set if new (B) negative; reset otherwise
Z: Set if new (B) 0; reset otherwise
H: Set if borrow from bit 4 of B; reset otherwise
P/V: Set if parity of new (B) is even; cleared otherwise
N: Set if bit 7 of data byte was 1; reset otherwise
C: Set if B 0 to FF; reset otherwise

Example: If HL contains 4200H, C contains 80H, B contains 10H, and memory location 4200 contains 81H, after execution of the instruction

OTDM

81H will have been written to output port 0080H, HL will contain 4201H, C will contain 81H, B will contain 0FH, and the flags will be S=Z=C=0, H=P/V=N=1.

OTIMR

Operation: (0, (C)) ← (HL)
HL ← HL + 1
C ← C + 1
B ← B - 1
Repeat above steps until B=0

Format:

1	1	1	0	1	1	0	1	ED
1	0	0	1	1	0	1	1	93

Description: A byte is read from memory at the address in HL, and is then written to the output port selected by placing the contents of C on A7-A0 and zeroes on A15-A8 (if any). Then the memory address in HL and the I/O address in C are both incremented, and a byte count in B is decremented. These steps are repeated until B has been decremented to 0. An initial B value of zero will cause 256 bytes to be transferred from memory to output ports. The flags are affected as shown below.

Interrupts can occur after any cycle, and the instruction will continue transparently upon return from the interrupt service routine. This instruction in useful for initializing a block of output registers, such as the 80180 "core" registers.

Timing:

	M Cycles	T States (180 reg)	T States (other 18x)
(new (B) nonzero)	4	16 (3, 3, 3, 3, 4 int)	17 (3, 3, 3, 4, 4 int)
(new (B) zero)	4	14 (3, 3, 3, 3, 2 int)	15 (3, 3, 3, 4, 2 int)

Condition Bits Affected:

S: Reset
Z: Set
H: Reset
P/V: Set
N: Set if bit 7 of last data byte was 1; reset otherwise
C: Reset

Example: If HL contains 4200H, C contains 80H, B contains 3, and memory locations 4200H to 4202H contains 13H 00H 81H, after execution of the instruction

OTDMR

13H will have been written to output port 0080H, 00 to port 0081H, 81H to port 0082H, HL will contain 4203H, C will contain 83H, and B will contain 0.

OTIR

Operation: (C) ← (HL), B ← B – 1, HL ← HL + 1

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	1	0	0	1	1	B3

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 read and temporarily stored in the CPU. Then, after the byte counter (B) is decremented, the contents of register C are placed on the bottom half (A7 through A0) of the address bus to select the I/O device at one of 256 possible ports. The decremented value of register B is placed on the top half (A15 through A8) 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 two and the instruction is repeated. If B has gone to zero, the instruction is terminated. Interrupts will be recognized and refresh cycles may be executed after each data transfer. Note that if B is set to zero prior to instruction execution, the instruction will output 256 bytes of data.

Note: This instruction is not particularly compatible with peripheral devices that decode A15-8 as part of a 16-bit I/O address.

Timing: If B ≠ 0:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	21 (4, 5, 3, 4, 5 int)	14 (3, 3, 3, 3, 2 int)	15 (3, 3, 3, 4, 2 int)

If B = 0:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	16 (4, 5, 3, 4)	12 (3, 3, 3, 3)	13 (3, 3, 3, 4)

Condition Bits Affected:

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

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

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

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:

```
51H (first)
A9H
03H (third)
```

OUT (C), r

Operation: (C) ← r

Format:

1	1	1	0	1	1	0	1	ED
0	1	←	r	→	0	0	1	

Description: The contents of register C are placed on the bottom half (A7 through A0) 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 (A15 through A8) of the address bus at this time. Then the byte contained in register r is placed on the data bus and written into the selected peripheral device. Register r identifies any of the CPU registers shown in the following table:

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

Timing:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	12 (4, 4, 4)	10 (3, 3, 1 int, 3)	11 (3, 3, 1 int, 4)

Condition Bits Affected:
None.

Example: If the contents of register C are 01H, the contents of register B are 02H, and the contents of register D are 5AH, after the execution of

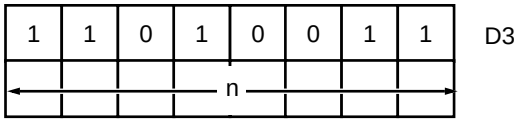
OUT (C),D

the byte 5AH will have been written to the peripheral device mapped to I/O port address 0201H. If the peripheral device ignores A15 through A8, the statement above should read "...the peripheral device mapped to I/O port address 01H."

OUT (n), A

Operation: (n) ← A

Format:



Description: The operand n is placed on the bottom half (A7 through A0) of the address bus to select the I/O device at one of 256 possible ports. The contents of the Accumulator (register A) appear on the top half (A15 through A8) 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.

Note: This instruction is not particularly compatible with peripherals that decode A15 through A8 as part of a 16-bit I/O address.

Timing:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
3	11 (4, 3, 4)	10 (3, 3, 1 int, 3)	11 (3, 3, 1 int, 4)

Condition Bits Affected:
None.

Example: If the contents of the Accumulator are 23H the execution of then after the execution of

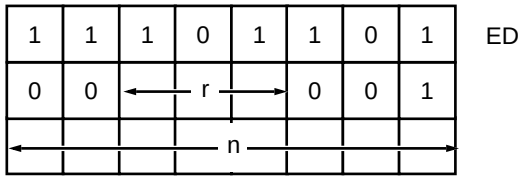
OUT (01H), A

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

OUT0 (n), r

Operation: (0, n) ← r

Format:



Description: The operand n is placed on A7 through A0 with zero on A15 through A8, to select an output port. The contents of the register selected by r, as shown in the following table, are then written to the selected output port (if any). The flags are not affected.

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

Timing:

M Cycles	T States (180 reg)	T States (other 18x)
4	13 (3, 3, 3, 3, 1 int)	14 (3, 3, 3, 4, 1 int)

Condition Bits Affected:
None

Example: If the contents of an 8018x processor's Input/Output Control Register (IOCR) are 1FH, the 16-bit I/O address of its ASCI0 Transmit Data Register is 0006H, and the contents of A are 41H, then after the execution of

```
OUT0  (6), A
```

the byte 41H will have been written to ASCI0 for serial transmission. Note also that under the same conditions, the execution of

```
OUT  (6), A
```

will **not** write the byte to ASCI0, rather to 16-bit I/O address 4106H!

OUTD

Operation: (C) ← (HL), B ← B – 1, HL ← HL – 1

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	0	1	0	1	1	AB

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 read and temporarily stored in the CPU. Then, after the byte counter (B) is decremented, the contents of register C are placed on the bottom half (A7 through A0) of the address bus to select the I/O device at one of 256 possible ports. The decremented value of register B 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.

Note: This instruction is not particularly compatible with peripheral devices that decode A15 through A8 as part of a 16-bit I/O address.

Timing:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	16 (4, 5, 3, 4)	12 (3, 3, 3, 3)	13 (3, 3, 3, 4)

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/O port address 07H.

OUTI

Operation: (C) ← (HL), B ← B – 1, HL ← HL + 1

Format:

1	1	1	0	1	1	0	1	ED
1	0	1	0	0	0	1	1	A3

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 (A7 through A0) of the address bus to select the I/O device at one of 256 possible ports. The decremented value of register B is placed on the top half (A15 through A8) 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.

Note: This instruction is not particularly compatible with peripheral devices that decode A15 through A8 as part of a 16-bit I/O address.

Timing:

M Cycles	Z80 T States	Z180 Register T States	Z18x Other Register T States
4	16 (4, 5, 3, 4)	12 (3, 3, 3, 3)	13 (3, 3, 3, 4)

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 thee 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 device mapped to I/O port address 07H.

POP pp

Operation: $pp_L \leftarrow (SP)$, $pp_H \leftarrow (SP+1)$, $SP \leftarrow SP+2$ **Format:**

POP qq (qq = BC, DE, HL, or AF)	1	1	z	z	0	0	0	1	
POP IX	1	1	0	1	1	1	0	1	DD
	1	1	1	0	0	0	0	1	E1
POP IY	1	1	1	1	1	1	0	1	FD
	1	1	1	0	0	0	0	1	E1

Description: Two bytes are "popped" from the stack in external memory, into 16-bit register pp. First a byte is read from memory at the address in the Stack Pointer SP, and is loaded into the less significant half of pp. Then SP is incremented by one, another byte is read from memory at the new value of SP, and is loaded into the more-significant half of pp. Finally SP is incremented again. No flags are affected unless the operand is AF. There is no checking for stack underflow.

"pp" can be AF (the Accumulator A as MS, flags as LS), any of the register pairs BC, DE, or HL, or an index register IX or IY. In the first form shown above, qq is encoded as follows:

Register	qq
BC	00
DE	01
HL	10
AF	11

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
POP qq	3	10 (4, 3, 3)	9 (3, 3, 3)
POP IX	4	14 (4, 4, 3, 3)	12 (3, 3, 3, 3)
POP IY	4	14 (4, 4, 3, 3)	12 (3, 3, 3, 3)

Condition Bits Affected:
Only if the operand is AF.

Example: If the SP contains 1000H, memory location 1000H contains 55H, and location 1001H contains 33H, then after the execution of

```
POP IX
```

IX will contain 3355H and SP will contain 1002H.

PUSH pp

Operation: (SP-1) ← pp_H, (SP-2) ← pp_L, SP ← SP-2

Format:

PUSH qq (qq = BC, DE, HL, or AF)	1	1	q	q	0	1	0	1	
PUSH IX	1	1	0	1	1	1	0	1	DD
	1	1	1	0	0	1	0	1	E5
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 16-bit register pp are "pushed" into the stack in external memory. First the contents of the Stack Pointer SP are decremented by one, and the contents of the more-significant byte of register pp are stored in memory at the new value of SP. Then SP is decremented again, and the contents of the less-significant byte of the register pp are stored in memory at the new value of SP. No flags are affected and there is no checking for stack overflow.

"pp" can be AF (the Accumulator A as MS, flags as LS), any of the register pairs BC, DE, or HL, or an index register IX or IY. In the first form shown above, qq is encoded as follows:

Register	qq
BC	00
DE	01
HL	10
AF	11

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
PUSH qq	3	11 (5, 3, 3)	11 (3, 2 int, 3, 3)
PUSH IX	4	15 (4, 4, 1 int, 3, 3)	14 (3, 3, 2 int, 3, 3)
PUSH IY	4	15 (4, 4, 1 int, 3, 3)	14 (3, 3, 2 int, 3, 3)

Condition Bits Affected:
None

Example: If the Accumulator A contains 22H, the flags contain 11H, and the Stack Pointer SP contains 1007H, then after the execution of

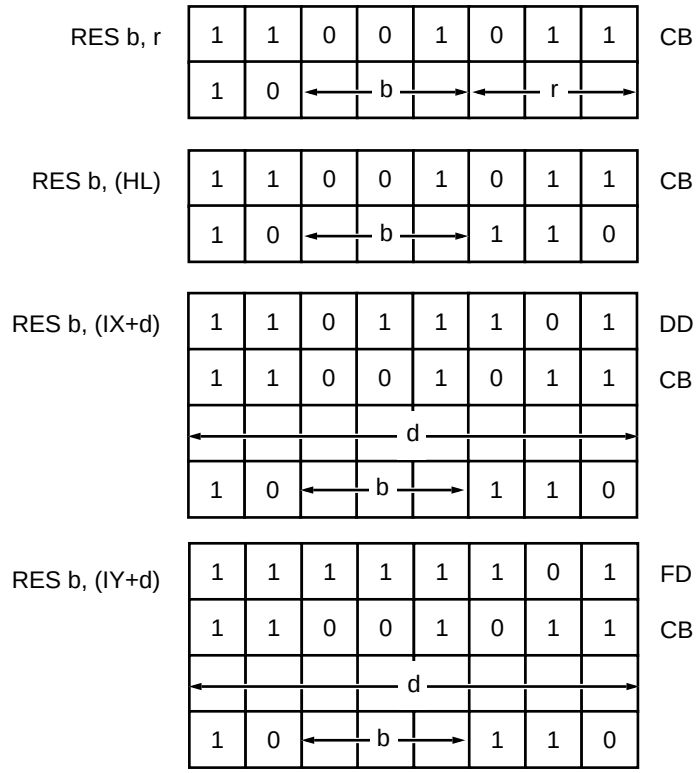
```
PUSH  AF
```

memory address 1006H will contain 22H, address 1005H will contain 11H, and SP will contain 1005H.

RES b, m

Operation: M ← m and not (2 ^b)

Format:



Description: Bit b in operand m is reset to 0. No flags are affected. b can be 0 for the LS bit through 7 for the MS bit. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
RES r	4	8 (4, 4)	7 (3, 3, 1 int)
RES (HL)	4	15 (4, 4, 4, 3)	12 (3, 3, 3, 1 int, 3)
RES (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)
RES (IY+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)

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).

RET

Operation: $PC_L \leftarrow (SP)$, $PC_H \leftarrow (SP+1)$, $SP \leftarrow (SP)+2$

Format:

1	1	0	0	1	0	0	1	C9

Description: The byte at the memory location specified by the contents of the Stack Pointer (SP) register pair are moved to the low order eight bits of the Program Counter (PC). The SP is now incremented and the byte at the memory location specified by the new contents of this instruction will be fetched from the memory location specified by the PC. This instruction is normally used to return to the calling sequence at the completion of a routine entered by a CALL instruction.

Timing:

M Cycles	Z80 T States	Z18x T States
3	10 (4, 3, 3)	9 (3, 3, 3)

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 of memory location 2001H are 18H, then after the execution of

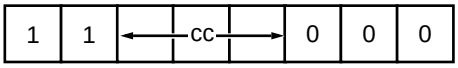
```
RET
```

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: $PC_L \leftarrow (SP)$, $PC_H \leftarrow (SP+1)$, $SP \leftarrow SP + 2$

Format:



Description: If condition cc is true, the byte at the memory location specified by the contents of the Stack Pointer (SP) is moved to the low order eight bits of the Program Counter (PC). The SP is then incremented and the byte at the memory location specified by the new contents of the SP are moved to the high order eight bits of the PC. The SP is then incremented again. The next opcode following this instruction will be fetched from the memory location specified by the PC. This instruction is normally used to return to the calling sequence at the completion of a routine entered by a CALL instruction. 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 values which correspond to condition bits in the Flag Register (register F):.

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

Timing: If cc is true:

M Cycles	Z80 T States	Z18x T States
3	11 (5, 3, 3)	10 (3, 1 int, 3, 3)

If cc is false:

M Cycles	Z80 T States	Z18x T States
1	5	5 (3, 2 int)

Condition Bits Affected:
None.

Example. If the S flag in the F register is set, 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 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:

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)

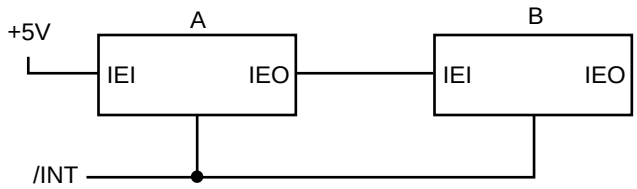
2. To signal a Z80 family I/O device that the interrupt routine has been completed. The RETI instruction facilitates the nesting of interrupts allowing higher priority devices to temporarily suspend service of lower priority service routines. Note: This instruction does not enable interrupts which were disabled when the interrupt routine was entered. Before doing the RETI instruction, the enable interrupt instruction (EI) should be executed to allow recognition of interrupts after completion of the RETI.

Timing:

Z80		Z185 (OMCR7 (M1E) = 1)		Z18x (Otherwise)	
M Cycles	**T States**	**M Cycles**	**T States**	**M Cycles**	**T States**
4	14 (4, 4, 3, 3)	4	13 (3, 3, 1 int, 3, 3))	6	22 (3, 3, 3 int, 3, 1 int, 3, 3, 3)

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, setting the IEO of A back to high. A second RETI is issued on completion of the B routine and the IEO of B is set back to high.

RETN

Operation: Return from non maskable interrupt

Format:

1	1	1	0	1	1	0	1	ED
0	1	0	0	0	1	0	1	45

Description: This instruction is used at the end of a non-maskable interrupt service routine to restore the contents of the Program Counter (PC) (analogous to the RET instruction). The state of IFF2 is copied back into IFF1 so that maskable interrupts are enabled immediately following the RETN if they were enabled before the non-maskable interrupt.

Timing:

M Cycles	Z80 T States	Z18x T States
4	14 (4, 4, 3, 3)	12 (3, 3, 3, 3)

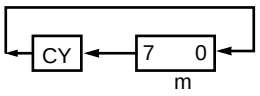
Condition Bits Affected:

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 restart at 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 a 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 again of 1000H. The program flow continues where it left off with an opcode fetch to address 1A45H.

RL m

Operation:



Format:

RL r	1	1	0	0	1	0	1	1	CB
	0	0	0	1	0	← r →			
RL (HL)	1	1	0	0	1	0	1	1	CB
	0	0	0	1	0	1	1	0	16
RL (IX+d)	1	1	0	1	1	1	0	1	DD
	1	1	0	0	1	0	1	1	CB
	← d →								
	0	0	0	1	0	1	1	0	16
RL (IY+d)	1	1	1	1	1	1	0	1	FD
	1	1	0	0	1	0	1	1	CB
	← d →								
	0	0	0	1	0	1	1	0	16

Description: The contents of the m operand are rotated left one bit position. The content of bit 7 is copied into the Carry flag and the previous content of the Carry flag is copied into bit 0, other flags being set as described below. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
RL r	2	8 (4, 4)	7 (3, 3, 1 int)
RL (HL)	4	15 (4, 4, 4, 3)	13 (3, 3, 3, 1 int, 3)
RL (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 30
RL (IY+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)

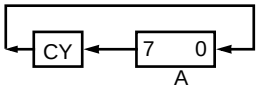
Condition Bits Affected:

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 Carry flag is 0 and register D contains 8FH, then after the execution of RL D the contents of register D and the Carry flag is 1 and register D contains IEH

RLA

Operation :



Format:

0	0	0	1	0	1	1	1	17

Description: The contents of the Accumulator (register A) are rotated left one bit position through the Carry flag. The previous content of the Carry flag is copied into bit 0.

Timing:

M Cycles	Z80 T States	Z18x T States
1	4	3

Condition Bits Affected :

S: Not affected
Z: Not affected
H: Reset
P/V: Not affected
N: Reset
C: Data from bit 7 of Accumulator

Example: If the contents of the Accumulator and the Carry flag are

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

after the execution of

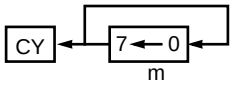
RLA

the contents of the Accumulator and the Carry flag will be

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

RLC m

Operation:



Format:

RLC r	1	1	0	0	1	0	1	1	CB
	0	0	0	0	0	r			
RLC (HL)	1	1	0	0	1	0	1	1	CB
	0	0	0	0	0	1	1	0	06
RLC (IX+d)	1	1	0	1	1	1	0	1	DD
	1	1	0	0	1	0	1	1	CB
	d								
	0	0	0	0	0	1	1	0	06
RLC (IY+d)	1	1	1	1	1	1	0	1	FD
	1	1	0	0	1	0	1	1	CB
	d								
	0	0	0	0	0	1	1	0	06

Description: The contents of the m operand are rotated left by one bit, the former bit 7 being copied into both bit 0 and the C flag, other flags being set as described below. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
RLC r	2	8 (4, 4)	7 (3, 3, 1 int)
RLC (HL)	4	15 (4, 4, 4, 3)	13 (3, 3, 3, 1 int, 3)
RLC (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)
RLC (IY+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)

Condition Bits Affected:

S: Set if new bit 7 is 1, reset otherwise
Z: Set if bits 7-0 are all 0; reset otherwise
H: Reset
P/V: Set if parity even; reset otherwise
N: Reset
C: Former bit 7 (same as new bit 0)

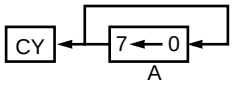
Example: If IY contains 1000H, and memory location 0FFEH contains 88H, then after the execution of

```
INC  (IY - 2)
```

the Carry flag will be 1 and memory location 0FFEH will contain11H.

RLCA

Operation:



Format:

0	0	0	0	0	1	1	1	07

Description: The contents of the Accumulator (register A) are rotated left one bit position. The sign bit (bit 7) is copied into the Carry flag and also into bit 0. Bit 0 is the least significant bit.

Timing:

M Cycles	Z80 T States	Z18x T States
1	4	3

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Not affected
N: Reset
C: Data from bit 7 of Accumulator

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

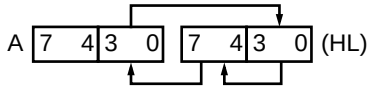
RLCA

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:

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.

Timing:

M Cycles	Z80 T States	Z180x T States
5	18 (4, 4, 3, 4 int, 3)	16 (3, 3, 3, 4 int, 3)

Condition Bits Affected:

S: Set if Accumulator is negative after operation; reset otherwise
Z: Set if Accumulator is zero after operation; reset otherwise
H: Reset
P/V: Set if parity of Accumulator 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

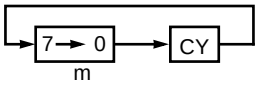
7	6	5	4	3	2	1	0	
0	1	1	1	1	0	1	0	Accumulator
0	0	1	1	0	0	0	1	(5000H)

after the execution of RLD the contents of the Accumulator and memory location 5000H will be

7	6	5	4	3	2	1	0	
0	1	1	1	0	0	1	1	Accumulator
0	0	0	1	1	0	1	1	(5000H)

RR m

Operation:



Format:

RR r	1	1	0	0	1	0	1	1	CB
	0	0	0	1	1	r			
RR (HL)	1	1	0	0	1	0	1	1	CB
	0	0	0	1	1	1	1	0	1E
RR (IX+d)	1	1	0	1	1	1	0	1	DD
	1	1	0	0	1	0	1	1	CB
	d								
	0	0	0	1	1	1	1	0	1E
RR (IY+d)	1	1	1	1	1	1	0	1	FD
	1	1	0	0	1	0	1	1	CB
	d								
	0	0	0	1	1	1	1	0	1E

Description: The contents of operand m are rotated right one bit position through the Carry flag. The content of bit 0 is copied into the Carry flag and the previous content of the Carry flag is copied into bit 7. Other flags are set as described below. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
RR r	2	8 (4, 4)	7 (3, 3, 1 int)
RR (HL)	4	15 (4, 4, 4, 3)	13 (3, 3, 3, 1 int, 3)
RR (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)
RR (IY+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)

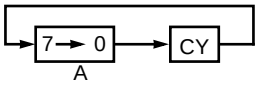
Condition Bits Affected:

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 of source register

Example: If the contents of the HL register pair are 4343H, and the contents of memory location 4343H contains DDH, and the Carry flag is 0, then after the execution of RR (HL), location 4343H contains 6EH and the Carry flag is 1.

RRA

Operation:



Format:

0	0	0	1	1	1	1	1	1F

Description: The contents of the Accumulator (register A) are rotated right one bit position through the Carry flag. The previous content of the Carry flag is copied into bit 7, and the previous content of bit 0 is copied into the Carry flag.

Timing:

M Cycles	Z80 T States	Z18x T States
1	4	3

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Not affected
N: Reset
C: Data from bit 0 of Accumulator

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

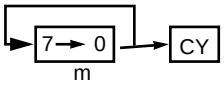
RRA

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

RRC m

Operation:



Format:

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

Description: The contents of the m operand are rotated right one bit position. The content of bit 0 is copied into the Carry flag and also into bit 7, other flags being set as described below. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
RRC r	2	8 (4, 4)	7 (3, 3, 1 int)
RRC (HL)	4	15 (4, 4, 4, 3)	13 (3, 3, 3, 1 int, 3)
RRC (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)
RRC (IY+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)

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 of source register

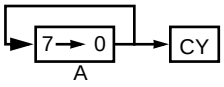
Example: If the contents of register A contains 31H, after the execution of

RRC A

A will contain 98H and the Carry flag will be 1.

RRCA

Operation:



Format:

0	0	0	0	1	1	1	1	0F

Description: The contents of the Accumulator (register A) are rotated right one bit position. Bit 0 is copied into the Carry flag and also into bit 7.

Timing:

M Cycles	Z80 T States	Z18x T States
1	4	3

Condition Bits Affected:

S: Not affected
Z: Not affected
H: Reset
P/V: Not affected
N: Reset
C: Data from bit 0 of Accumulator

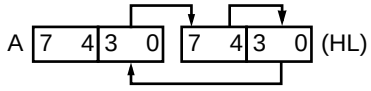
Example: If the Accumulator contains 11H,after the execution of

RRCA

the Accumulator will contain 88H and the Carry flag will be 1.

RRD

Operation:



Format:

1	1	1	0	1	1	0	1	ED
0	1	1	0	0	1	1	1	67

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.

Timing: t

M Cycles	Z80 T States	Z18x T States
4	18 (4, 4, 3, 4 int, 3)	16 (3, 3, 3, 4 int, 3)

Condition Bits Affected:

S: Set if Accumulator is negative after operation; reset otherwise
Z: Set if Accumulator. is zero after operation; reset otherwise
H: Reset
P/V: Set if parity of Accumulator 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

7	6	5	4	3	2	1	0	
1	0	0	0	0	1	0	0	Accumulator
0	0	1	0	0	0	0	0	(5000H)

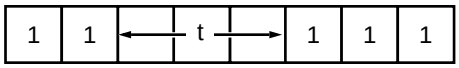
after the execution of RRD, Accumulator and memory location 5000H will contain

7	6	5	4	3	2	1	0	
1	0	0	0	0	0	0	0	Accumulator
0	1	0	0	0	0	1	0	(5000H)

RST p

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

Format:



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. .

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

Timing:

M Cycles	Z80 T States	Z18x T States
3	11 (5, 3, 3)	11 (3, 2 int, 3, 3)

Condition Bits Affected:
None.

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

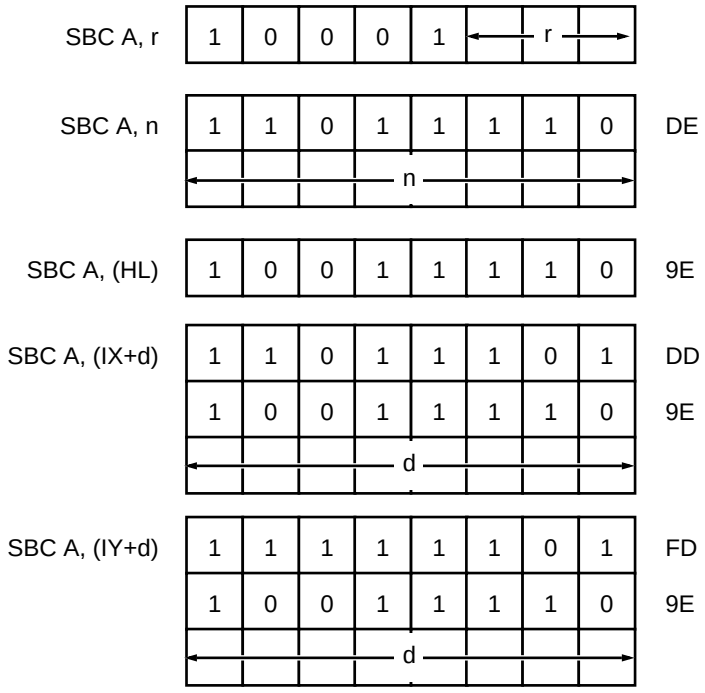
RST 18H (Object code 1101111)

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

SBC A, s

Operation: A ← A – s – CY

Format:



Description: The s operand and the Carry flag are subtracted from the contents of the Accumulator A, the result is stored in A, and the flags are set as described below. "s" can be any of a register r, an "immediate" value n in the instruction itself, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects a source register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
SBC A, r	1	4	4 (3 + 1 int)
SBC A, n	2	7 (4, 3)	6 (3, 3)
SBC A, (HL)	2	7 (4, 3)	6 (3, 3)
SBC A, (IX+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)
SBC A, (IY+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Set if borrow from bit 4; reset otherwise
P/V: Set if overflow; reset otherwise
N: Set
C: Set if borrow; 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:

1	1	1	0	1	1	0	1	ED
0	1	s	s	0	0	1	0	

Description: The contents of the register pair ss (any of register pairs BC, DE, HL, or SP) and the Carry Flag are subtracted from the contents of register pair HL, the result is stored in HL, and the Carry flag reflects whether a borrow was needed. Operand ss is specified as follows in the assembled object code.

Register Pair	ss
BC	00
DE	01
HL	10
SP	11

Timing:

M Cycles	Z80 T States	Z18x T States
2	15 (4, 4, 7 int)	10 (3, 3, 4 int)

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Set if a borrow from bit 12; reset otherwise
P/V: Set if overflow; reset otherwise
N: Set
C: Set if borrow; 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:

0	0	1	1	0	1	1	1	37

Description: The Carry flag in the F register is set.

Timing:

M Cycles	Z80 T States	Z18x T States
1	4	3

Condition Bits Affected:

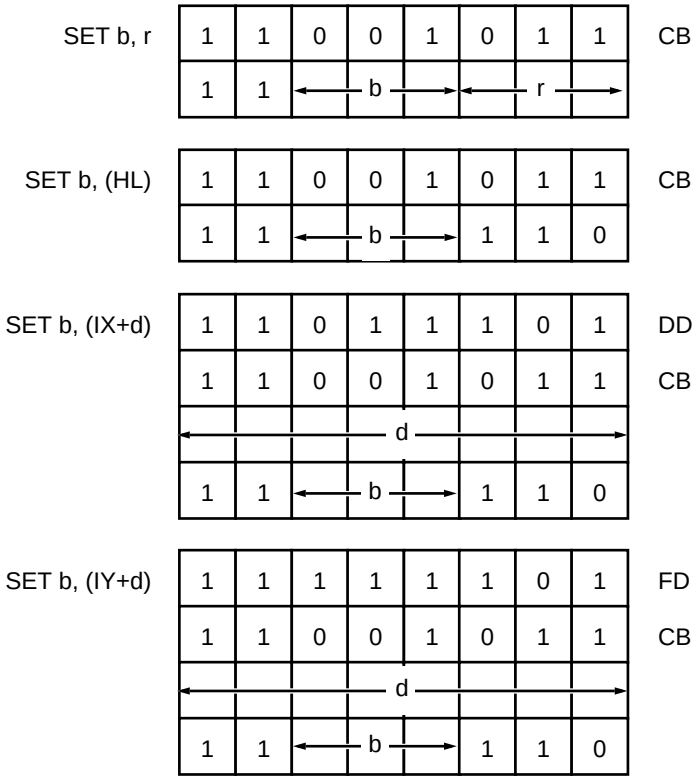
S: Not affected
Z: Not affected
H: Reset
P/V: Not affected
N: Reset
C: Set

Note: There is no specific "Clear Carry Flag" instruction. Use OR A, A to clear the Carry flag.

SET b, m

Operation: m ← m OR (2 ^ b)

Format:



Description: Bit b of operand m is set to 1. No flags are affected. b can be 0 for the LS bit through 7 for the MS bit. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
SET b, r	2	8 (4, 4)	7 (3, 3, 1 int)
SET b, (HL)	4	15 (4, 4, 4, 3)	13 (3, 3, 3, 1 int, 3)
SET b, (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)
SET b, (IY+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)

Condition Bits Affected:
None

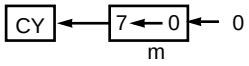
Example: If HL contains 4567H, and memory location 4567H contains 33H, then after the execution of

```
SET  7, (HL)
```

memory location 4567H will contain B3H.

SLA m

Operation:



Format:

SLA r	1	1	0	0	1	0	1	1	CB
	0	0	1	0	0	r			
SLA (HL)	1	1	0	0	1	0	1	1	CB
	0	0	1	0	0	1	1	0	26
SLA (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	0	1	1	0	26
SLA (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	0	1	1	0	26

Description: The operand m is arithmetically shifted left one bit position. The content of bit 7 is copied into the Carry flag, other flags being set as described below. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX and IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
SLA r	2	8 (4, 4)	7 (3, 3, 1 int)
SLA (HL)	4	15 (4, 4, 4, 3)	13 (3, 3, 3, 1 int, 3)
SLA (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)
SLA (IY+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)

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 register L contains BIH, then after the execution of

SLA L

the contents of register L will contain 62H and the Carry flag will be set.

SLP

Operation: Enter Sleep or System Stop mode

Format:

1	1	1	0	1	1	0	1	ED
0	1	1	1	0	1	1	0	76

Description: If the IOSTOP bit in the 8018x processor's I/O Control Register is 0, this instruction places the device in Sleep mode. Sleep mode is "deeper" than Halt mode (see HALT instruction) in that the internal CPU clock stops, the DMA channels will not operate, DRAM refresh (if any) stops, the Address bus is 3-stated, and control signals are driven high except that they are 3-stated while BUSACK is Low. Like Halt mode, Sleep mode can terminated by a Reset or by an interrupt request from an external or internal source, including the ASCIs, PRTs, and CSI/O. More details on Sleep mode and exit therefrom are given in the section dealing with this mode.

If the IOSTOP bit in the IOCR is 1 when this instruction is executed, we say that the device enters System Stop mode. This mode differs from Sleep mode only in that, since the ASCIs, PRTs, and CSI/O are stopped, they cannot generate an interrupt to terminate the mode.

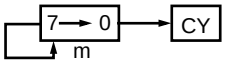
Timing:

M Cycles	18x T States
2	Indefinite: 8 (3, 3, 2 int) minimum

Condition Bits Affected:
None

SRA m

Operation:



Format:

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

Description: The operand m is arithmetically shifted right one bit position. The content of bit 0 is copied into the Carry flag and the previous contents of bit 7 is unchanged. Other flags are set as described below. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
SRA r	2	8 (4, 4)	7 (3, 3, 1 int)
SRA (HL)	4	15 (4, 4, 4, 3)	13 (3, 3, 3, 1 int, 3)
SRA (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)
SRA (IY+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)

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

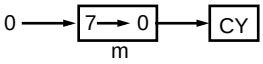
Example: If the contents of the Index Register IX are 1000H, and the contents of memory location 1003H are B8H, then after the execution of

SRA (IX+3)

memory location 1003H will contain DCH and the Carry flag will be 0.

SRL m

Operation:



Format:

SRL r	1	1	0	0	1	0	1	1	CB
	0	0	1	1	1	r			
SRL (HL)	1	1	0	0	1	0	1	1	CB
	0	0	1	1	1	1	1	0	3E
SRL (IX+d)	1	1	0	1	1	1	0	1	DD
	1	1	0	0	1	0	1	1	CB
	d								
	0	0	1	1	1	1	1	0	3E
SRL (IY+d)	1	1	1	1	1	1	0	1	FD
	1	1	0	0	1	0	1	1	CB
	d								
	0	0	1	1	1	1	1	0	3E

Description: The contents of operand m are shifted right one bit position. The content of bit 0 is copied into the Carry flag, bit 7 is reset, and other flags are set as described below. "m" can be a register r, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects the register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
SRL r	2	8 (4, 4)	7 (3, 3, 1 int)
SRL (HL)	4	15 (4, 4, 4, 3)	13 (3, 3, 3, 1 int, 3)
SRL (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)
SRL (IY+d)	6	23 (4, 4, 3, 5, 4, 3)	19 (3, 3, 3, 3, 3, 1 int, 3)

Condition Bits Affected:

S: Reset
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 register B contains 8FH, then after the execution of

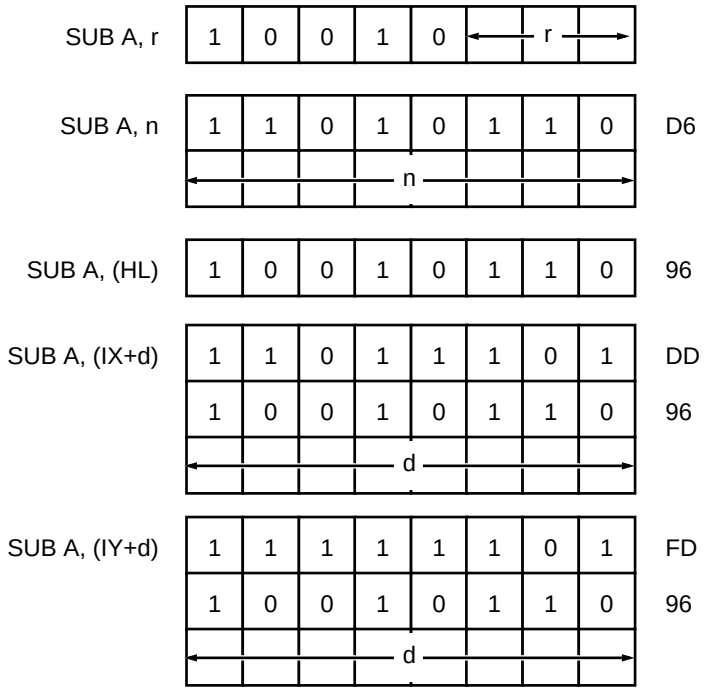
SRL B

register B will contain 47H and the Carry flag will be set.

SUB A, s

Operation: A ← A – s

Format:



Description: The s operand is subtracted from the contents of Accumulator A, the result is stored in A, and the flags are set as described below. "s" can be any of a register r, an "immediate" value n in the instruction itself, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects a source register as follows:

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

Timing:

Instruction	M Cycle	Z80 T States	Z18x T States
SUB A, r	1	4	4 (3 + 1 int)
SUB A, n	2	7 (4, 3)	6 (3, 3)
SUB A, (HL)	2	7 (4, 3)	6 (3, 3)
SUB A, (IX+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)
SUB A, (IY+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)

Condition Bits Affected:

S: Set if result is negative; reset otherwise
Z: Set if result is zero; reset otherwise
H: Set if borrow from bit 4; reset otherwise
P/V: Set if overflow; reset otherwise
N: Set
C: Set if borrow; reset otherwise

Example: If the Accumulator contents are 29H, and register D contains 11H, after the execution of

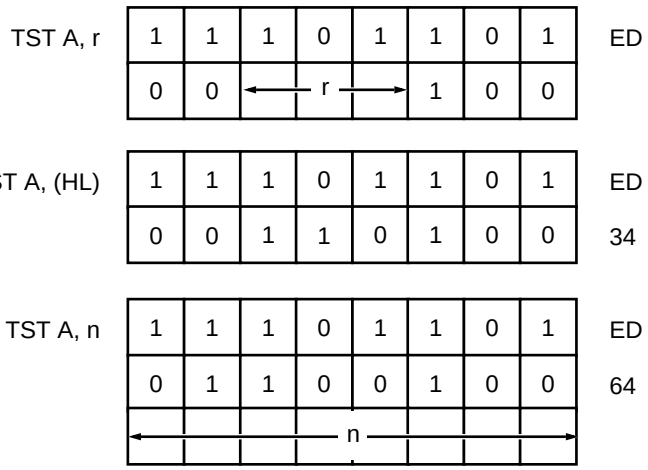
```
SUB D
```

the Accumulator will contain 18H.

TST A, s

Operation: Test Accumulator (A AND s)

Format:



Description: The contents of the A register are logically ANDed with the s operand, which can be a register, the memory location pointed to by HL, or an immediate value. The result is discarded, but the flags are set to reflect the result as shown below. For the register case the r field of the instruction is encoded as follows:

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

Timing:

Instruction	M Cycles	18x T States
TST A, r	2	7 (3, 3, 1 int)
TST A, (HL)	3	10 (3, 3, 3, 1 int)
TST A, n	3	9 (3, 3, 3)

Condition Bits Affected:

S: Set if there are 1s in bit 7 of A and the operand, reset otherwise
Z: Set if all 8 bits of the result are 0, reset otherwise
H: Set
P/V: set if the parity of the result is even; reset otherwise
N: Reset
C: Reset

Example: If A contains 83H and B contains 7FH, then the instruction

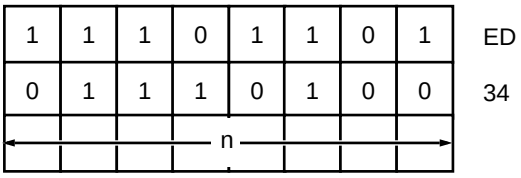
```
TST  A, B
```

produces a (discarded) AND result of 03H, and the flags are set as S=0, Z=0, and P/V=1.

TSTIO n

Operation: Set flags per (C/ AND n)

Format:



Description: The contents of the C register are placed on A7-A0 as an I/O port address, with zeroes on A15-A8. The contents of the input port selected by this address (if any) are read and logically ANDed with the immediate data operand n. The result is discarded, but the flags are set to reflect the result as shown below.

Timing:

M Cycles	T States (180 register)	T States (18x other reg)
3	12 (3, 3, 3, 3 int)	13 (3, 3, 4, 3 int)

Condition Bits Affected:

S: Set if there are 1s in bit 7 of both the input byte and the operand, reset otherwise
Z: Set if all 8 bits of the result are 0, reset otherwise
H: Set
P/V: Set if the parity of the result is even; reset otherwise
N: Reset
C: Reset

Example: If the contents of C are 45H and input port 0045H contains 3AH, then the instruction

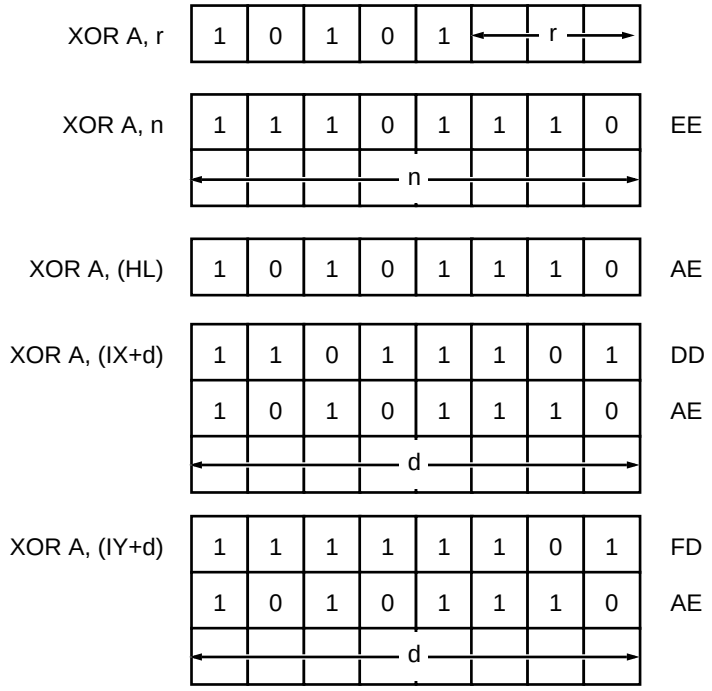
```
TSTIO  88H
```

produces a (discarded) AND result of 08H, and the flags are set as S=0, Z=0, and P/V=0.

XOR A, s

Operation: A ← A XOR s

Format:



Description: The s operand is logically exclusive ORed with the contents of the Accumulator A, the result is stored in A, and the flags are set as described below. "s" can be any of a register r, an "immediate" value n in the instruction itself, a memory location selected by the contents of the HL register pair, or a memory location selected by the sum of the contents of an index register IX or IY and a signed 8-bit displacement d. In the register form, r selects a source register as follows:

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

Timing:

Instruction	M Cycles	Z80 T States	Z18x T States
XOR A, r	1	4	4 (3 + 1 int)
XOR A, n	2	7 (4, 3)	6 (3, 3)
XOR A, (HL)	2	7 (4, 3)	6 (3, 3)
XOR A, (IX+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)
XOR A, (IY+d)	4	19 (4, 4, 3, 5 int, 3)	14 (3, 3, 3, 2 int, 3)

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: Reset

Example: If the Accumulator contains 96H, then after the execution of

XORA, 5DH

the Accumulator will contain CBH

Zilog, Inc. 210 East Hacienda Ave.
Campbell, CA 95008-6600
Telephone (408) 370-8000
FAX 408 370-8056
Internet: http://www.zilog.com