Z80 Instruction Set

Z80 Assembly Language

The assembly language allows the user to write a program without concern for memory addresses or machine instruction formats. It uses symbolic addresses to identify memory locations and mnemonic codes (Op Codes and operands) to represent the instructions. Labels (symbols) are assigned to a particular instruction step in a source program to identify that step as an entry point for use in subsequent instructions. Operands following each instruction represent storage locations, registers, or constant values. The assembly language also includes assembler directives that supplement the machine instruction. A pseudo-op, for example, is a statement that is not translated to a machine instruction, but rather is interpreted as a directive that controls the assembly process.

A program written in assembly language is called a source program, which consists of symbolic commands called statements. Each statement is written on a single line and may consist of from one to four entries: A label field, an operation field, an operand field, and a comment field. The source program is processed by the assembler to obtain a machine language program (object program) that can be executed directly by the Z80 CPU.

ZiLOG provides several assemblers that differ in the features offered. Both absolute and relocatable assemblers are available with the Development and Micro-computer Systems. The absolute assembler is contained in base level software operating in a 16K memory space, while the relocating assembler is part of the RIO environment operating in a 32K memory space.

Z80 Status Indicator Flags

The flag registers (F and F') supply information to the user about the status of the Z80 at any given time. The bit positions for each flag is listed below:

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

Symbol	Field Name
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 flag registers contains 6 bits of status information that are set or cleared by CPU operations. (Bits 3 and 5 are not used.) Four of these bits (C, P/V, Z, and S) may be tested for use with conditional JUMP, CALL, or RETURN instructions. Two flags may not be tested (H, N) and are used for BCD arithmetic.

Carry Flag

The Carry Flag (C) is set or cleared depending on the operation performed. For ADD instructions that generate a Carry, and SUB instructions that generate a Borrow, the Carry Flag sets. The Carry Flag is reset by an ADD instruction that does not generate a Carry, and by a SUB instruction that does not generate a Borrow. This saved Carry facilitates software routines

for extended precision arithmetic. Also, the DAA instruction sets the Carry Flag if the conditions for making the decimal adjustment are met.

For instructions RLA, RRA, RLS, and RRS, the Carry bit is used as a link between the least significant byte (LSB) and most significant byte (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 is reset.

The Carry Flag can also be set by the Set Carry Flag (SCF) and complemented by the Compliment Carry Flag (CCF) instructions.

Add/Subtract Flag

The Add/Subtract Flag (N) is used by the Decimal Adjust Accumulator instruction (DAA) to distinguish between ADD and SUB instructions. For ADD instructions, N is cleared to 0. For SUB instructions, N is set to 1.

Add/Subtract Flag

The Decimal Adjust Accumulator instruction (DAA) uses this flag to distinguish between ADD and SUBTRACT instructions. For all ADD instructions, N sets to 0. For all SUBTRACT instructions, N sets to 1.

Parity/Overflow Flag (P/V)

This flag is set to a specific state depending on the operation 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 is determined by examining the sign bits of the operands.

For addition, operands with different signs 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   ADDEND
+105    =   0110   1001   AUGEND
-------------------------------
+225    =   1110   0001   (-95)      SUM
```

The two numbers added together resulted in a number that exceeds +127 and the two positive operands have 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 signs 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 set.

Another method for identifying an Overflow is to observe the Carry to 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 resulting parity is Even. The number of 1 bits in a byte are counted. If the total is Odd, ODD parity is flagged (P = 0). 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 decrements to 0, the flag is cleared 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 value of the interrupt enable flip-flop (IFF2) for storage or testing.

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

Half Carry Flag

The Half-Carry Flag (H) is set (1) or cleared (0) 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 cleared (0) according to the following table:

H Flag	Add	Subtract
1	A Carry occurs from Bit 3 to Bit 4	A Borrow from Bit 4 occurs
0	No Carry occurs from Bit 3 to Bit 4	No Borrow from Bit 4 occurs

Zero Flag

The Zero Flag (Z) is set (1) or cleared (0) if the result generated by the execution of certain instructions is 0.

For 8-bit arithmetic and logical operations, the Z flag is set to a 1 if the resulting byte in the Accumulator is 0. If the byte is not 0, 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 indicated by the value 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 0, the Z flag is 1, otherwise the Z flag is 0. Also for byte inputs from I/O devices using IN r, (C), the Z flag is set to indicate a 0-byte input.

Sign Flag

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, the binary twos-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 twos 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 using an IN r, (C) instruction, the S Flag indicates either positive (S = 0) or negative (S = 1) data.

Z80 Instruction Description

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

M Cycles: 2T States: 7(4,3) 4 MHzE.T.: 1.75

indicates that the instruction consists of 2 machine cycles. The first cycle contains 4 clock periods (T States). The second cycle contains 3 clock periods for a total of 7 clock periods or T States. The instruction executes in 1.75 microseconds.

Register format is indicated for each instruction with the most-significant bit to the left and the least-significant bit to the right.

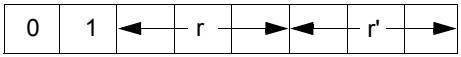
8-Bit Load Group

LD r, r'

Operation: r, ← r'

Op Code: LD

Operands: r, r'



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

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

M Cycles	T States	MHz E.T.
1	4	1.0

Condition Bits Affected: None

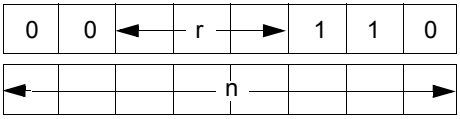
Example: If the H register contains the number 8AH, and the E register contains 10H, the instruction LD H, E results in both registers containing 10H.

LD r,n

Operation: r ← n

Op Code: LD

Operands: r, n



Description: The 8-bit integer n is loaded to any register r, where r identifies register A, B, C, D, E, H, or L, assembled as follows in the object code:

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

M Cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected: None

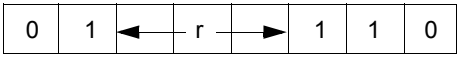
Example: At execution of `LD E, A5H` the contents of register E are `A5H`.

LD r, (HL)

Operation: r ← (HL)

Op Code: LD

Operands: r, (HL)



Description: The 8-bit contents of memory location (HL) are loaded to register r, where r identifies register A, B, C, D, E, H, or L, assembled as follows in the object code:

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

M Cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected: None

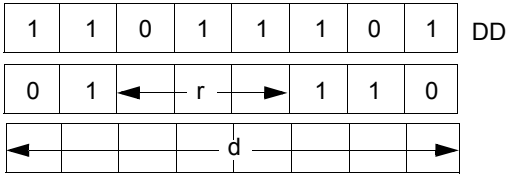
Example: If register pair HL contains the number 75A1H, and memory address 75A1H contains byte 58H, the execution of LD C, (HL) results in 58H in register C.

LD r, (IX+d)

Operation: r ← (IX+d)

Op Code: LD

Operands: r, (IX+d)



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

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

M Cycles	T States	4 MHz E.T.
5	19 (4, 4, 3, 5, 3)	2.50

Condition Bits Affected: None

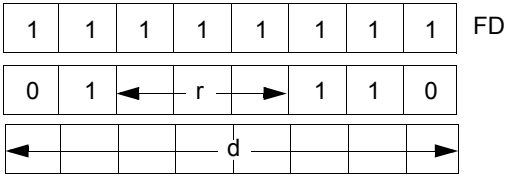
Example: If the Index Register IX contains the number 25AFH, the instruction LD B, (IX+19H) causes the calculation of the sum 25AFH + 19H, which points to memory location 25C8H. If this address contains byte 39H, the instruction results in register B also containing 39H.

LD r, (IY+d)

Operation: r ← (IY+D)

Op Code: LD

Operands: r, (IY+d)



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

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

M Cycles	T States	4 MHz E.T.
5	19 (4, 4, 3, 5, 3)	4.75

Condition Bits Affected: None

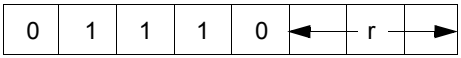
Example: If the Index Register IY contains the number 25AFH, the instruction `LD B, (IY+19H)` causes the calculation of the sum `25AFH + 19H`, which points to memory location `25C8H`. If this address contains byte `39H`, the instruction results in register B also containing `39H`.

LD (HL), r

Operation: (HL) ← r

Op Code: LD

Operands: (HL), r



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

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

M Cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected: None

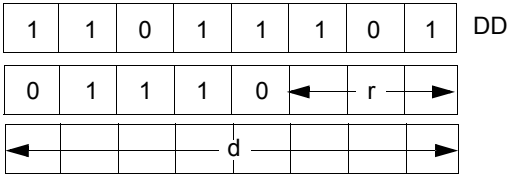
Example: If the contents of register pair HL specifies memory location `2146H`, and the B register contains byte `29H`, at execution of `LD (HL), B` memory address `2146H` also contains `29H`.

LD (IX+d), r

Operation: (IX+d) ← r

Op Code: LD

Operands: (IX+d), r



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

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

M Cycles	T States	4 MHz E.T.
5	19 (4, 4, 3, 5, 3)	4.75

Condition Bits Affected: None

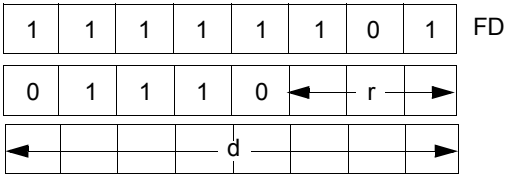
Example: If the C register contains byte 1CH, and the Index Register IX contains 3100H, then the instruction LID (IX+6H),C performs the sum 3100H + 6H and loads 1CH to memory location 3106H.

LD (IY+d), r

Operation: (IY+d) ← r

Op Code: LD

Operands: (IY+d), r



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

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

M Cycles	T States	4 MHz E.T.
5	19 (4, 4, 3, 5, 3)	4.75

Condition Bits Affected: None

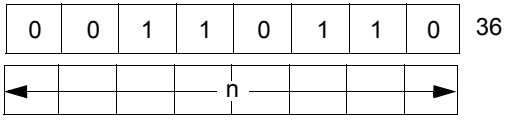
Example: If the C register contains byte 48H, and the Index Register IY contains 2A11H, then the instruction LD (IY+4H), C performs the sum 2A11H + 4H, and loads 48H to memory location 2A15.

LD (HL), n

Operation: (HL) ← n

Op Code: LD

Operands: (HL), n



Description: Integer n is loaded to the memory address specified by the contents of the HL register pair.

M Cycles	T States	4 MHz E.T.
3	10 (4, 3, 3)	2.50

Condition Bits Affected: None

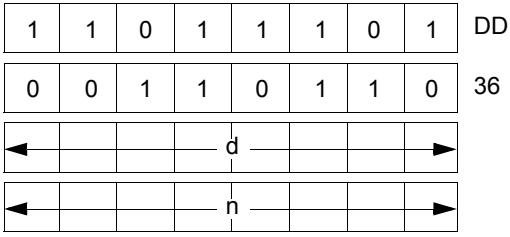
Example: If the HL register pair contains 4444H, the instruction `LD (HL), 28H` results in the memory location 4444H containing byte 28H.

LD (IX+d), n

Operation: (IX+d) ← n

Op Code: LD

Operands: (IX+d), n



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

M Cycles	T States	4 MHz E.T.
5	19 (4, 4, 3,5,3)	4.75

Condition Bits Affected: None

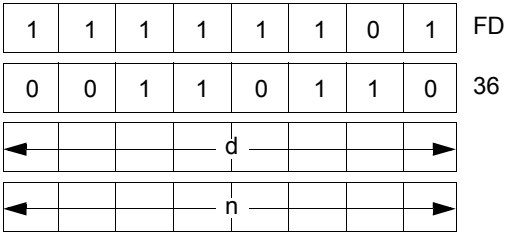
Example: If the Index Register IX contains the number 219AH, the instruction LD (IX+5H), 5AH results in byte 5AH in the memory address 219FH.

LD (IY+d), n

Operation: (IY+d) ← n

Op Code: LD

Operands: (IY+d), n



Description: Integer n is loaded to the memory location specified by the contents of the Index Register summed with the two's complement displacement integer d.

M Cycles	T States	4 MHz E.T.
5	19 (4, 4, 3, 5, 3)	2.50

Condition Bits Affected: None

Example: If the Index Register IY contains the number `A940H`, the instruction `LD (IY+10H),97H` results in byte `97H` in memory location `A950H`.

LD A, (BC)

Operation: A ← (BC)

Op Code: LD

Operands: A, (BC)

0	0	0	0	1	0	1	0	0A

Description: The contents of the memory location specified by the contents of the BC register pair are loaded to the Accumulator.

M Cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected: None

Example: If the BC register pair contains the number `4747H`, and memory address `4747H` contains byte `12H`, then the instruction `LD A, (BC)` results in byte `12H` in register A.

LD A, (DE)

Operation: A ← (DE)

Op Code: LD

Operands: A, (DE)

0	0	0	1	1	0	1	0	1A

Description: The contents of the memory location specified by the register pair DE are loaded to the Accumulator.

M Cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected: None

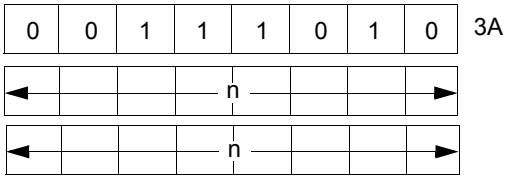
Example: If the DE register pair contains the number `30A2H` and memory address `30A2H` contains byte `22H`, then the instruction `LD A, (DE)` results in byte `22H` in register A.

LD A, (nn)

Operation: A ← (nn)

Op Code: LD

Operands: A, (nn)



Description: The contents of the memory location specified by the operands `nn` are loaded to the Accumulator. The first `n` operand after the Op Code is the low order byte of a 2-byte memory address.

M Cycles	T States	4 MHz E.T.
4	13 (4, 3, 3, 3)	3.25

Condition Bits Affected: None

Example: If the contents of `nn` is number `8832H`, and the content of memory address `8832H` is byte `04H`, at instruction `LD A, (nn)` byte `04H` is in the Accumulator.

LD (BC), A

Operation: (BC) ← A

Op Code: LD

Operands: (BC), A

0	0	0	0	0	0	1	0	02

Description: The contents of the Accumulator are loaded to the memory location specified by the contents of the register pair BC.

M Cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected: None

Example: If the Accumulator contains `7AH` and the BC register pair contains `1212H` the instruction `LD (BC), A` results in `7AH` in memory location `1212H`.

LD (DE), A

Operation: (DE) ← A

Op Code: LD

Operands: (DE), A

0	0	0	1	0	0	1	0	12

Description: The contents of the Accumulator are loaded to the memory location specified by the contents of the DE register pair.

M cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected: None

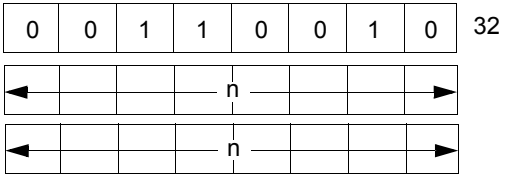
Example: If the contents of register pair DE are `1128H`, and the Accumulator contains byte `A0H`, the instruction `LD (DE), A` results in `A0H` in memory location `1128H`.

LD (nn), A

Operation: (nn) ← A

Op Code: LD

Operands: (nn), A



Description: The contents of the Accumulator are loaded to the memory address specified by the operand `nn`. The first `n` operand after the Op Code is the low order byte of `nn`.

M Cycles	T States	4 MHz E.T.
4	13 (4, 3, 3, 3)	3.25

Condition Bits Affected: None

Example: If the contents of the Accumulator are byte `D7H`, at execution of `LD (3141 H), AD7H` results in memory location `3141H`.

LD A, I

Operation: A ← I

Op Code: LD

Operands: A, I

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 to the Accumulator.

M Cycles	T States	MHz E.T.
2	9 (4, 5)	2.25

Condition Bits Affected:

S is set if I-Register is negative; reset otherwise

Z is set if I-Register is zero; reset otherwise

H is reset

P/V contains contents of IFF2

N is reset

C is not affected

If an interrupt occurs during execution of this instruction, the Parity flag contains a 0.

LD A, R

Operation: A, ← R

Op Code: LD

Operands: A, R

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 to the Accumulator.

M Cycles	T States	MHz E.T.
2	9 (4, 5)	2.25

Condition Bits Affected:

S is set if, R-Register is negative; reset otherwise

Z is set if R-Register is zero; reset otherwise

H is reset

P/V contains contents of IFF2

N is reset

C is not affected

If an interrupt occurs during execution of this instruction, the parity flag contains a 0.

LD I,A

Operation: I ← A

Op Code: LD

Operands: I, A

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 to the Interrupt Control Vector Register, I.

M Cycles	T States	MHz E.T.
2	9 (4, 5)	2.25

Condition Bits Affected: None

LD R, A

Operation: R ← A

Op Code: LD

Operands: R, A

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 to the Memory Refresh register R.

M Cycles	T States	MHz E.T.
2	9 (4, 5)	2.25

Condition Bits Affected: None

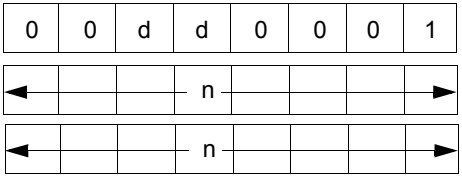
16-Bit Load Group

LD dd, nn

Operation: dd ← nn

Op Code: LD

Operands: dd, nn



Description: The 2-byte integer `nn` is loaded to the `dd` register pair, where dd defines the BC, DE, HL, or SP register pairs, assembled as follows in the object code:

Pair	dd
BC	`00`
DE	`01`
HL	`10`
SP	`11`

The first `n` operand after the Op Code is the low order byte.

M Cycles	T States	4 MHz E.T.
2	10 (4, 3, 3)	2.50

Condition Bits Affected: None

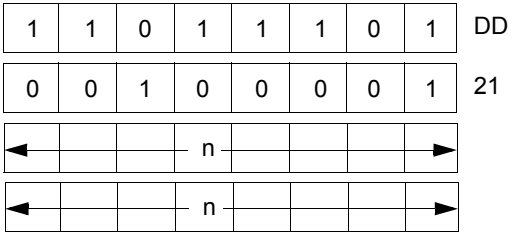
Example: At execution of `LD HL, 5000H` the contents of the HL register pair is `5000H`.

LD IX, nn

Operation: Ix ← nn

Op Code: LD

Operands: IX, nn



Description: Integer `nn` is loaded to the Index Register IX. The first `n` operand after the Op Code is the low order byte.

M Cycles	T States	4 MHz E.T.
4	14 (4, 4, 3, 3)	3.50

Condition Bits Affected: None

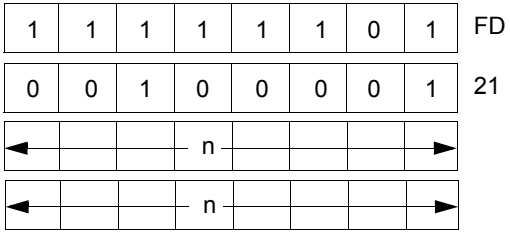
Example: At instruction `LD IX, 45A2H` the Index Register contains integer `45A2H`.

LD IY, nn

Operation: IY ← nn

Op Code: LD

Operands: IY, nn



Description: Integer nn is loaded to the Index Register IY. The first n operand after the Op Code is the low order byte.

M Cycles	T States	4 MHz E.T.
4	14 (4, 4, 3, 3)	3.50

Condition Bits Affected: None

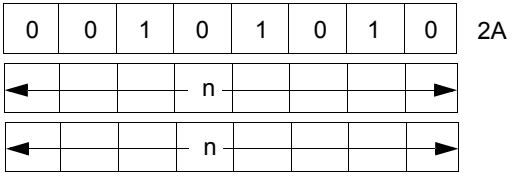
Example: At instruction `LD IY, 7733H` the Index Register IY contains the integer `7733H`.

LD HL, (nn)

Operation: H ← (nn+1), L ← (nn)

Op Code: LD

Operands: HL, (nn)



Description: The contents of memory address (nn) are loaded to the low order portion of register pair HL (register L), and the contents of the next highest memory address (nn+1) are loaded to the high order portion of HL (register H). The first n operand after the Op Code is the low order byte of nn.

M Cycles	T States	4 MHz E.T.
5	16 (4, 3, 3, 3, 3)	4.00

Condition Bits Affected: None

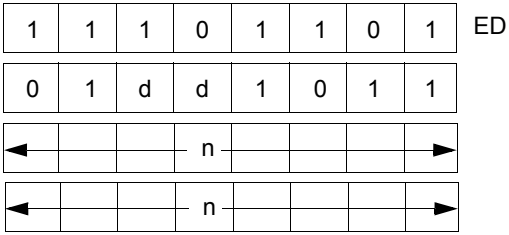
Example: If address 4545H contains 37H, and address 4546H contains A1H, at instruction LD HL, (4545H) the HL register pair contains A137H.

LD dd, (nn)

Operation: ddh ← (nn+1) ddl ← (nn)

Op Code: LD

Operands: dd, (nn)



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

Pair	dd
BC	00
DE	01
HL	10
SP	11

The first n operand after the Op Code is the low order byte of (nn).

M Cycles	T States	4 MHz E.T.
6	20 (4, 4, 3, 3, 3, 3)	5.00

Condition Bits Affected: None

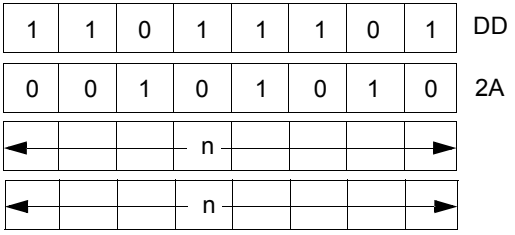
Example: If Address 2130H contains 65H, and address 2131M contains 78H, at instruction LD BC, (2130H) the BC register pair contains 7865H.

LD IX, (nn)

Operation: IXh ← (nn+1), IXI ← (nn)

Op Code: LD

Operands: IX, (nn)



Description: The contents of the address (nn) are loaded to the low order portion of Index Register IX, and the contents of the next highest memory address (nn+1) are loaded to the high order portion of IX. The first n operand after the Op Code is the low order byte of nn.

M Cycles	T States	4 MHz E.T.
6	20 (4, 4, 3, 3, 3, 3)	5.00

Condition Bits Affected: None

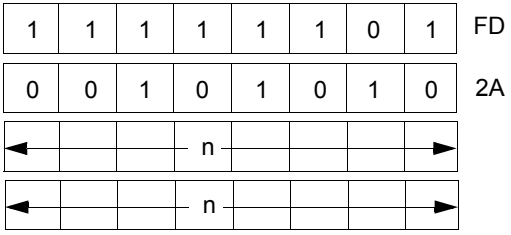
Example: If address 6666H contains 92H, and address 6667H contains DAH, at instruction LD IX, (6666H) the Index Register IX contains DA92H.

LD IY, (nn)

Operation: IYh ← (nn+1), IYI ← nn)

Op Code: LD

Operands: IY, (nn)



Description: The contents of address (nn) are loaded to the low order portion of Index Register IY, and the contents of the next highest memory address (nn+1) are loaded to the high order portion of IY. The first n operand after the Op Code is the low order byte of nn.

M Cycles	T States	4 MHz E.T.
6	20 (4, 4, 3, 3, 3, 3)	5.00

Condition Bits Affected: None

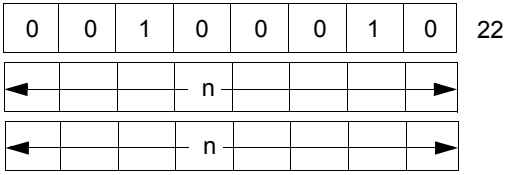
Example: If address 6666H contains 92H, and address 6667H contains DAH, at instruction LD IY, (6666H) the Index Register IY contains DA92H.

LD (nn), HL

Operation: (nn+1) ← H, (nn) ← L

Op Code: LD

Operands: (nn), HL



Description: The contents of the low order portion of register pair HL (register L) are loaded to memory address (nn), and the contents of the high order portion of HL (register H) are loaded to the next highest memory address (nn+1). The first n operand after the Op Code is the low order byte of nn.

M Cycles	T States	4 MHz E.T.
5	16 (4, 3, 3, 3, 3)	4.00

Condition Bits Affected: None

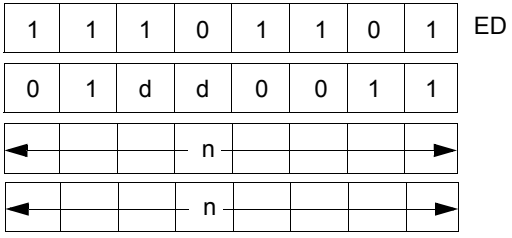
Example: If the content of register pair HL is 483AH, at instruction LD (B2291-1), HL address B229H contains 3AH, and address B22AH contains 48H.

LD (nn), dd

Operation: (nn+1) ← ddh, (nn) ← ddl

Op Code: LD

Operands: (nn), dd



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

Pair	dd
BC	00
DE	01
HL	10
SP	11

The first n operand after the Op Code is the low order byte of a two byte memory address.

M Cycles	T States	4 MHz E.T.
6	20 (4, 4, 3, 3, 3, 3)	5.00

Condition Bits Affected: None

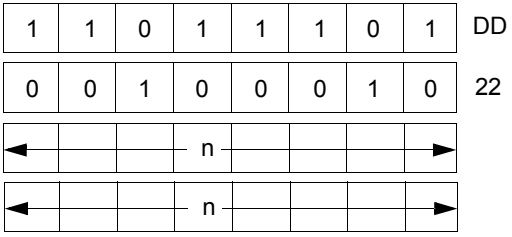
Example: If register pair BC contains the number 4644H, the instruction LD (1000H), BC results in 44H in memory location 1000H, and 46H in memory location 1001H.

LD (nn), IX

Operation: (nn+1) ← IXh, (nn) ← IXI

Op Code: LD

Operands: (nn), IX



Description: The low order byte in Index Register IX is loaded to memory address (nn); the upper order byte is loaded to the next highest address (nn+1). The first n operand after the Op Code is the low order byte of nn.

M Cycles	T States	4 MHz E.T.
6	20 (4, 4, 3, 3, 3, 3)	5.00

Condition Bits Affected: None

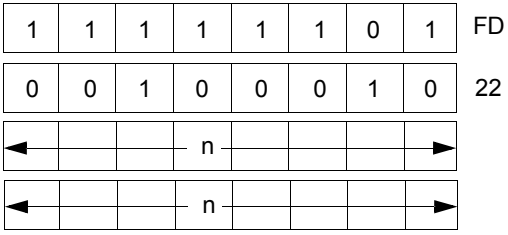
Example: If the Index Register IX contains 5A30H, at instruction LD (4392H), IX memory location 4392H contains number 30H, and location 4393H contains 5AH.

LD (nn), IY

Operation: (nn+1) ← IYh, (nn) ← IYI

Op Code: LD

Operands: (nn), IY



Description: The low order byte in Index Register IY is loaded to memory address (nn); the upper order byte is loaded to memory location (nn+1). The first n operand after the Op Code is the low order byte of nn.

M Cycles	T States	4 MHz E.T.
6	20 (4, 4, 3, 3, 3, 3)	5.00

Condition Bits Affected: None

Example: If the Index Register IY contains 4174H at instruction LD (8838H), IY memory location 8838H contains number 74H, and memory location 8839H contains 41H.

LD SP, HL

Operation: SP ← HL

Op Code: LD

Operands: SP, HL

1	1	1	r	1	0	0	1	F9

Description: The contents of the register pair HL are loaded to the Stack Pointer (SP).

M Cycles	T States	4 MHz E.T.
1	6	1.5

Condition Bits Affected: None

Example: If the register pair HL contains `442EH`, at instruction `LD SP, HL` the Stack Pointer also contains `442EH`.

LD SP, IX

Operation: SP ← - IX

Op Code: LD

Operands: SP, 1X

1	1	0	1	1	1	0	1	DD
1	1	1	1	1	0	0	1	F9

Description: The 2-byte contents of Index Register IX are loaded to the Stack Pointer (SP).

M Cycles	T States	4 MHz E.T.
2	10 (4, 6)	2.50

Condition Bits Affected: None

Example: If the contents of the Index Register IX are `98DAH`, at instruction `LD SP, IX` the contents of the Stack Pointer are also `98DAH`.

LD SP, IY

Operation: SP ← IY

Op Code: LD

Operands: SP, IY

1	1	1	1	1	1	0	1	FD
1	1	1	1	1	0	0	1	F9

Description: The 2-byte contents of Index Register IY are loaded to the Stack Pointer SP.

M Cycles	T States	4 MHz E.T.
2	10 (4, 6)	2.50

Condition Bits Affected: None

Example: If Index Register IY contains the integer `A227H`, at instruction `LD SP, IY` the Stack Pointer also contains `A227H`.

PUSH qq

Operation: (SP-2) ← qqL, (SP-1) ← qqH

Op Code: PUSH

Operands: qq

1	1	q	q	0	1	0	1

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

Pair	qq
BC	00
DE	01
HL	10
AF	11

M Cycles	T States	4 MHz E.T.
3	11 (5, 3, 3)	2.75

Condition Bits Affected: None

Example: If the AF register pair contains 2233H and the Stack Pointer contains 1007H, at instruction PUSH AF memory address 1006H contains 22H, memory address 1005H contains 33H, and the Stack Pointer contains 1005H.

PUSH IX

Operation: (SP-2) ← IXL, (SP-1) ← IXH

Op Code: PUSH

Operands: IX

1	1	0	1	1	1	0	1	DD
1	1	1	0	0	1	0	1	E5

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

M Cycles	T States	4 MHz E.T.
4	15 (4, 5, 3, 3)	3.75

Condition Bits Affected: None

Example: If the Index Register IX contains `2233H` and the Stack Pointer contains `1007H`, at instruction `PUSH IX` memory address `1006H` contains `22H`, memory address `1005H` contains `33H`, and the Stack Pointer contains `1005H`.

PUSH IY

Operation: (SP-2) ← IYL, (SP-1) ← IYH

Op Code: PUSH

Operands: IY

1	1	1	1	1	1	0	1	FD
1	1	1	0	0	1	0	1	E5

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

M Cycles	T States	4 MHz E.T.
4	15 (4, 5, 3, 3)	3.75

Condition Bits Affected: None

Example: If the Index Register IY contains `2233H` and the Stack Pointer Contains `1007H`, at instruction `PUSH IY` memory address `1006H` contains `22H`, memory address `1005H` contains `33H`, and the Stack Pointer contains 1005H.

POP qq

Operation: qqH ← (SP+1), qqL ← (SP)

Op Code: POP

Operands: qq

1	1	q	q	0	0	0	1

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

Pair	r
BC	00
DE	01
HL	10
AF	11

M Cycles	T States	4 MHz E.T.
3	10 (4, 3, 3)	2.50

Condition Bits Affected: None

Example: If the Stack Pointer contains 1000H, memory location 1000H contains 55H, and location 1001H contains 33H, the instruction POP HL results in register pair HL containing 3355H, and the Stack Pointer containing 1002H.

POP IX

Operation: IXH ← (SP+1), IXL ← (SP)

Op Code: POP

Operands: IX

1	1	0	1	1	1	0	1	DD
1	1	1	0	0	0	0	1	E1

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

M Cycles	T States	4 MHz E.T.
4	14 (4, 4, 3, 3)	3.50

Condition Bits Affected: None

Example: If the Stack Pointer contains `1000H`, memory location `1000H` contains `55H`, and location `1001H` contains `33H`, the instruction `POP IX` results in Index Register IX containing `3355H`, and the Stack Pointer containing `1002H`.

POP IY

Operation: IYH ← (SP-X1), IYL ← (SP)

Op Code: POP

Operands: IY

1	1	0	1	1	1	0	1	DD
1	1	1	1	1	1	0	1	FD

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

M Cycles	T States	4 MHz E.T.
4	14 (4, 4, 3, 3)	3.50

Condition Bits Affected: None

Example: If the Stack Pointer Contains `1000H`, memory location `1000H` contains `55H`, and location `1001H` contains `33H`, the instruction `POP IY` results in Index Register IY containing `3355H`, and the Stack Pointer containing `1002H`.

Exchange, Block Transfer, and Search Group

EX DE, HL

Operation: DE ↔ HL

Op Code: EX

Operands: DE, HL

1	1	1	0	1	0	1	1	EB

Description: The 2-byte contents of register pairs DE and HL are exchanged.

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected: None

Example: If the content of register pair DE is the number `2822H`, and the content of the register pair HL is number `499AH`, at instruction `EX DE, HL` the content of register pair DE is `499AH`, and the content of register pair HL is `2822H`.

EX AF, AF'

Operation: AF ↔ AF'

Op Code: EX

Operands: AF, AF'

0	0	0	0	1	0	0	0	08

Description: The 2-byte contents of the register pairs AF and AF are exchanged.

Register pair AF consists of registers A' and F'.

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected: None

Example: If the content of register pair AF is number `9900H`, and the content of register pair AF is number `5944H`, at instruction `EX AF, AF'` the contents of AF is `5944H`, and the contents of AF' is `9900H`.

EXX

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

Op Code: EXX

Operands: —

1	1	0	1	1	0	0	0	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.

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected: None

Example: If the contents of register pairs BC, DE, and HL are the numbers `445AH`, `3DA2H`, and `8859H`, respectively, and the contents of register pairs BC', DE', and HL' are `0988H`, `9300H`, and `00E7H`, respectively, at instruction `EXX` the contents of the register pairs are as follows: BC' contains 0988H; DE' contains `9300H`; HL contains `00E7H`; BC' contains `445AH`; DE' contains `3DA2H`; and HL' contains `8859H`.

EX (SP), HL

Operation: H ↔ (SP+1), L ↔ (SP)

Op Code: EX

Operands: (SP), HL

1	1	1	0	0	0	1	1	E3

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

M Cycles	T States	4 MHz E.T.
5	19 (4, 3, 4, 3, 5)	4.75

Condition Bits Affected: None

Example: If the HL register pair contains `7012H`, the SP register pair contains `8856H`, the memory location `8856H` contains byte `11H`, and memory location `8857H` contains byte `22H`, then the instruction `EX (SP), HL` results in the HL register pair containing number `2211H`, memory location `8856H` containing byte `12H`, memory location `8857H` containing byte `70H` and Stack Pointer containing `8856H`.

EX (SP), IX

Operation: IXH ↔ (SP+1), IXL ↔ (SP)

Op Code: EX

Operands: (SP), IX

1	1	0	1	1	1	0	1	DD
1	1	1	0	0	0	1	1	E3

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

M cycles	T States	4 MHz E.T.
6	23 (4, 4, 3, 4, 3, 5)	5.75

Condition Bits Affected: None

Example: If the Index Register IX contains `3988H`, the SP register pair Contains `0100H`, memory location `0100H` contains byte `90H`, and memory location `0101H` contains byte `48H`, then the instruction `EX (SP), IX` results in the IX register pair containing number `4890H`, memory location `0100H` containing `88H`, memory location `0101H` containing `39H`, and the Stack Pointer containing `0100H`.

EX (SP), IY

Operation: IYH ↔ (SP+1), IYL ↔ (SP)

Op Code: EX

Operands: (SP), IY

1	1	1	1	1	1	0	1	FD
1	1	1	0	0	0	1	1	E3

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

M Cycles	T States	4 MHz E.T.
6	23 (4, 4, 3, 4, 3, 5)	5.75

Condition Bits Affected: None

Example: If the Index Register IY contains `3988H`, the SP register pair contains `0100H`, memory location `0100H` contains byte `90H`, and memory location `0101H` contains byte `48H`, then the instruction `EX (SP), IY` results in the IY register pair containing number `4890H`, memory location `0100H` containing `88H`, memory location `0101H` containing `39H`, and the Stack Pointer containing `0100H`.

LDI

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

Op Code: LDI

Operands: (SP), HL

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 and the BC (Byte Counter) register pair is decremented.

M Cycles	T States	4 MHz E.T.
4	16 (4, 4, 3, 5)	4.00

Condition Bits Affected:

S is not affected
Z is not affected
H is reset
P/V is set if BC -1 ≠ 0; reset otherwise
N is reset
C is not affected

Example: If the HL register pair contains `1111H`, memory location `1111H` contains byte `88H`, the DE register pair contains `2222H`, the memory location `2222H` contains byte `66H`, and the BC register pair contains `7H`, then the instruction `LDI` results in the following contents in register pairs and memory addresses:

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

LDIR

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

Op Code: LDIR

Operands: B8

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

Description: This 2-byte instruction transfers a byte of data from the memory location addressed by the contents of the HL register pair to the memory location addressed by the DE register pair. 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 are recognized and two refresh cycles are executed after each data transfer. When BC is set to zero prior to instruction execution, the instruction loops through 64 Kbytes.

For BC $\neq$ 0:

M Cycles	T States	4 MHz E.T.
5	21 (4, 4, 3, 5, 5)	5.25

For BC = 0:

M Cycles	T States	4 MHz E.T.
4	16 (4, 4, 3, 5)	4.00

Condition Bits Affected:

S is not affected
Z is not affected
H is reset
P/V is reset
N is reset
C is not affected

Example: If the HL register pair contains 11111H, the DE register pair contains 2222H, the BC register pair contains 0003H, and memory locations have these contents:

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

then at execution of LDIR the contents of register pairs and memory locations are:

HL	contains 1114H		
DE	contains 2225H		
BC	contains 0000H		
(1111H)	contains 88H	(2222H)	contains 88H
(1112H)	contains 36H	(2223H)	contains 36H
(1113H)	contains A5H	(2224H)	contains A5H

LDD

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

Op Code: LDD

Operands: —

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

Description: This 2-byte instruction transfers a byte of data from the memory location addressed by the contents of the HL register pair to the memory location addressed by the contents of the DE register pair. Then both of these register pairs including the BC (Byte Counter) register pair are decremented.

M Cycles	T States	4 MHz E.T.
4	16 (4, 4, 3, 5)	4.00

Condition Bits Affected:

S is not affected
Z is not affected
H is reset
P/V is set if BC -1 ≠ 0; reset otherwise
N is reset
C is not affected

Example: If the HL register pair contains `1111H`, memory location `1111H` contains byte `88H`, the DE register pair contains `2222H`, memory location `2222H` contains byte `66H`, and the BC register pair contains `7H`, then instruction `LDD` results in the following contents in register pairs and memory addresses:

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

LDDR

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

Op Code: LDDR

Operands: —

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

Description: This 2-byte instruction transfers a byte of data from the memory location addressed by the contents of the HL register pair to the memory location addressed by the contents of the DE register pair. Then both of these 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 are recognized and two refresh cycles execute after each data transfer.

When BC is set to zero, prior to instruction execution, the instruction loops through 64 Kbytes.

For BC $\neq$ 0:

M Cycles	T States	4 MHz E.T.
5	21 (4, 4, 3, 5, 5)	5.25

For BC = 0:

M Cycles	T States	4 MHz E.T.
4	16 (4, 4, 3, 5)	4.00

Condition Bits Affected:

S is not affected
Z is not affected
H is reset
P/V is reset
N is 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)	contains	A5H	(2225H)	contains	C5H
(1113H)	contains	36H	(2224H)	contains	59H
(1112H)	contains	88H	(2223H)	contains	66H

Then at execution of LDDR the contents of register pairs and memory locations are:

HL	contains	1111H			
DE	contains	2222H			
DC	contains	0000H			
(1114H)	contains	A5H	(2225H)	contains	A5H
(1113H)	contains	36H	(2224H)	contains	36H
(1112H)	contains	88H	(2223H)	contains	88H

CPI

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

Op Code: CPI

Operands: —

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

M Cycles	T States	4 MHz E.T.
4	16 (4, 4, 3, 5)	4.00

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if A is (HL); reset otherwise
H is set if borrow from bit 4; reset otherwise
P/V is set if BC -1 is not 0; reset otherwise
N is set
C is not affected

Example: If the HL register pair contains `1111H`, memory location `1111H` contains `3BH`, the Accumulator contains `3BH`, and the Byte Counter contains `0001H`. At execution of `CPI` the Byte Counter contains `0000H`, the HL register pair contains `1112H`, the Z flag in the F register sets, and the P/V flag in the F register resets. There is no effect on the contents of the Accumulator or address `1111H`.

CPIR

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

Op Code: CPIR

Operands: —

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 a true compare, a condition bit is set. HL is incremented and the Byte Counter (register pair BC) is decremented. If decrementing causes BC to go to zero or if A = (HL), the instruction is terminated. If BC is not zero and A ≠ (HL), the program counter is decremented by two and the instruction is repeated. Interrupts are recognized and two refresh cycles are executed after each data transfer.

If BC is set to zero before instruction execution, the instruction loops through 64 Kbytes if no match is found.

For BC ≠ 0 and A ≠ (HL):

M cycles	T States	4 MHz E.T.
5	21 (4, 4, 3, 5, 5)	5.25

For BC = 0 and A = (HL):

M Cycles	T States	4 MHz E.T.
4	16 (4, 4, 3, 5)	4.00

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if A equals (HL); reset otherwise
H is set if borrow from bit 4; reset otherwise
P/V is set if BC -1 does not equal 0; reset otherwise
N is set
C is not affected

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

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

Then, at execution of CPIR the contents of register pair HL is 1114H, the contents of the Byte Counter is 0004H, the P/V flag in the F register sets, and the Z flag in the F register sets.

CPD

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

Op Code: CPD

Operands: —

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 a true compare, a condition bit is set. The HL and Byte Counter (register pair BC) are decremented.

M Cycles	T States	4 MHz E.T.
4	16 (4, 4, 3, 5)	4.00

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if A equals (HL); reset otherwise
H is set if borrow from bit 4; reset otherwise
P/V is set if BC -1 x 0; reset otherwise
N is set
C is not affected

Example: If the HL register pair contains `1111H`, memory location `1111H` contains `3BH`, the Accumulator contains `3BH`, and the Byte Counter contains `0001H`. At execution of `CPD` the Byte Counter contains `0000H`, the HL register pair contains `1110H`, the flag in the F register sets, and the P/V flag in the F register resets. There is no effect on the contents of the Accumulator or address `1111H`.

CPDR

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

Op Code: CPDR

Operands: —

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 a true compare, a condition bit is set. The HL and BC (Byte Counter) register pairs are decremented. If decrementing causes the BC to go to zero or if A = (HL), the instruction is terminated. If BC is not zero and A = (HL), the program counter is decremented by two and the instruction is repeated. Interrupts are recognized and two refresh cycles execute after each data transfer. When BC is set to zero, prior to instruction execution, the instruction loops through 64 Kbytes if no match is found.

For BC ≠ 0 and A ≠ (HL):

M Cycles	T States	4 MHz E.T.
5	21 (4, 4, 3, 5, 5)	5.25

For BC = 0 and A = (HL):

M Cycles	T States	4 MHz E.T.
4	16 (4, 4, 3, 5)	4.00

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if A = (HL); reset otherwise
H is set if borrow form bit 4; reset otherwise
P/V is set if BC -1 ≠ 0; reset otherwise
N is set
C is not affected

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

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

Then, at execution of CPDR the contents of register pair HL are 1115H, the contents of the Byte Counter are 0004H, the P/V flag in the F register sets, and the Z flag in the F register sets.

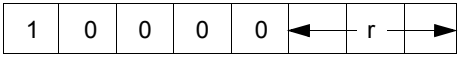
8-Bit Arithmetic Group

ADD A, r

Operation: A ← A + r

Op Code: ADD

Operands: A, r



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

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

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected:

S is set if result is negative; reset otherwise

Z is set if result is zero; reset otherwise

H is set if carry from bit 3; reset otherwise

P/V is set if overflow; reset otherwise

N is reset

C is set if carry from bit 7; reset otherwise

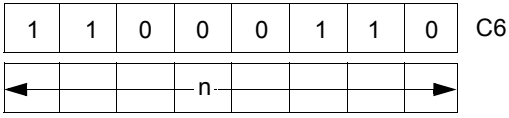
Example: If the contents of the Accumulator are `44H`, and the contents of register C are `11H`, at execution of `ADD A,C` the contents of the Accumulator are `55H`.

ADD A, n

Operation: A ← A + n

Op Code: ADD

Operands: A, n



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

M Cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected:

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

Example: If the contents of the Accumulator are `23H`, at execution of `ADD A, 33H` the contents of the Accumulator are `56H`.

ADD A, (HL)

Operation: A ← A + (HL)

Op Code: ADD

Operands: A, (HL)

1	0	0	0	0	1	1	0	86

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

M Cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected:

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

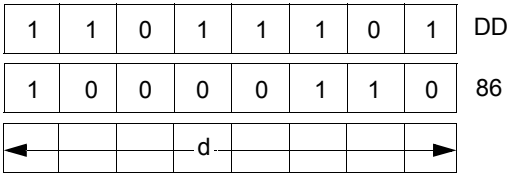
Example: If the contents of the Accumulator are `A0H`, and the content of the register pair HL is `2323H`, and memory location `2323H` contains byte `08H`, at execution of `ADD A, (HL)` the Accumulator contains `A8H`.

ADD A, (IX + d)

Operation: A ← A + (IX+d)

Op Code: ADD

Operands: A, (IX + d)



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

M Cycles	T States	4 MHz E.T.
5	19 (4, 4, 3, 5, 3)	4.75

Condition Bits Affected:

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

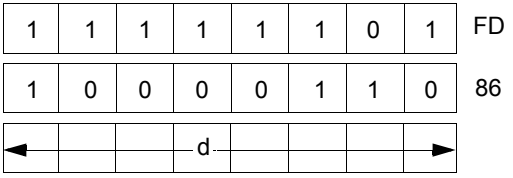
Example: If the Accumulator contents are `11H`, the Index Register IX contains 1000H, and if the contents of memory location `1005H` is `22H`, at execution of `ADD A, (IX + 5H)` the contents of the Accumulator are `33H`.

ADD A, (IY + d)

Operation: A ← A + (ID+d)

Op Code: ADD

Operands: A, (IY + d)



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

M Cycles	T States	4 MHz E.T.
5	19(4, 4, 3, 5, 3)	4.75

Condition Bits Affected:

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

Example: If the Accumulator contents are 11H, the Index Register Pair IY contains `1000H`, and if the content of memory location `1005H` is `22H`, at execution of `ADD A, (IY + 5H)` the contents of the Accumulator are `33H`.

ADC A, s

Operation: A ← A + s + CY

Op Code: ADC

Operands: A, s

This s operand is any of r, n, (HL), (IX+d), or (lY+d) as defined for the analogous ADD instruction. These possible Op Code/operand combinations are assembled as follows in the object code:

ADC A,r	1	0	0	0	1	◄	r*	►	
ADC A,n	1	1	0	0	1	1	1	0	CE
	◄			n				►	
ADC A, (HL)	1	0	0	0	1	1	1	0	8E
ADC A, (IX+d)	1	1	0	1	1	1	1	0	DD
	1	0	0	0	1	1	1	0	8E
	◄			d				►	
ADC A, (IY+d)	1	1	1	1	1	1	0	1	FD
	1	0	0	0	1	1	1	0	8E
	◄			d				►	

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

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

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

Instruction	M Cycle	T States	4 MHz E.T.
ADC A, r	1	4	1.00
ADC A, n	2	7 (4, 3)	1.75
ADC A, (HL)	2	7 (4, 3)	1.75
ADC A, (IX+d)	5	19 (4, 4, 3, 5, 3)	4.75
ADC A, (lY+d)	5	19 (4, 4, 3, 5, 3)	4.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is set if carry from bit 3; reset otherwise
P/V is set if overflow; reset otherwise
N is reset
C is 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, at execution of ADC A, (HL) the Accumulator contains 27H.

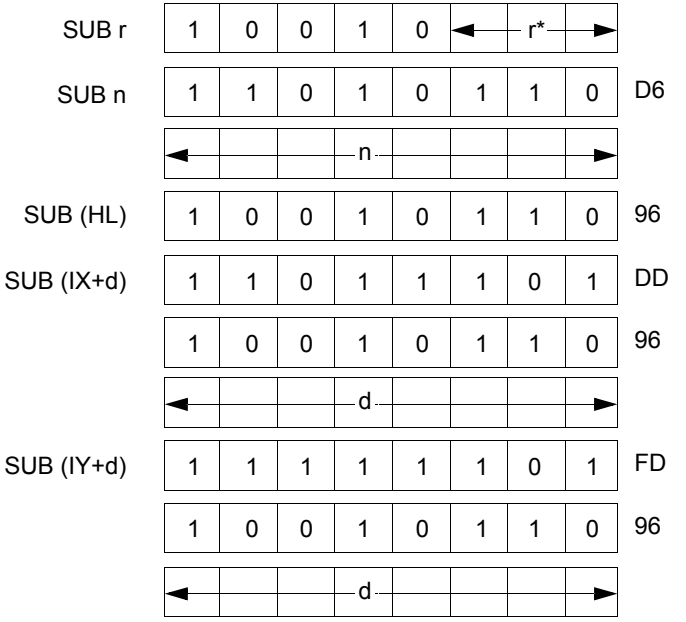
SUB s

Operation: A ← A - s

Op Code: SUB

Operands: s

This s operand is any of r, n, (HL), (IX+d), or (lY+d) as defined for the analogous ADD instruction. These possible Op Code/operand combinations are assembled as follows in the object code:



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

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

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

Instruction	M Cycle	T States	4 MHz E.T.
SUB r	1	4	1.00
SUB n	2	7 (4, 3)	1.75
SUB (HL)	2	7 (4, 3)	1.75
SUB (IX+d)	5	19 (4, 4, 3, 5, 3)	4.75
SUB (lY+d)	5	19 (4, 4, 3, 5, 3)	4.75

Condition Bits Affected:

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

Example: If the Accumulator contents are 29H, and register D contains 11H, at execution of `SUB D` the Accumulator contains 18H.

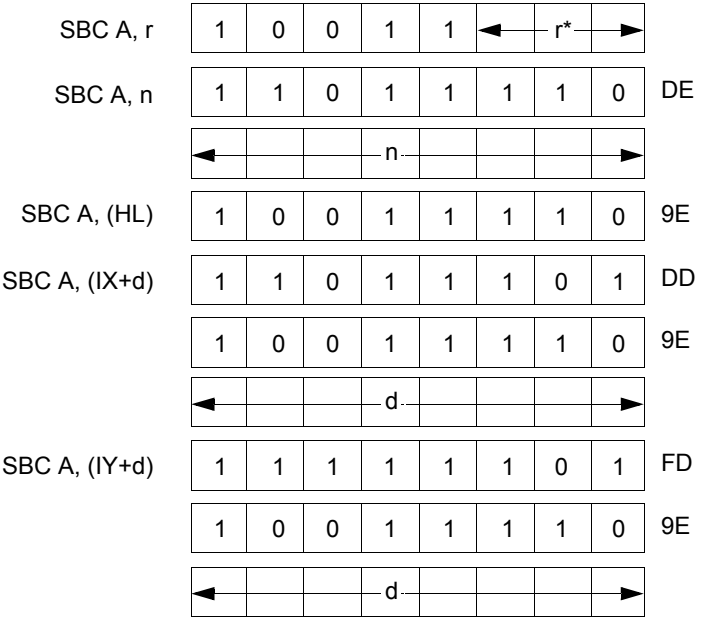
SBC A, s

Operation: A ← A - s - CY

Op Code: SBC

Operands: A, s

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



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

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

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

Instruction	M Cycles	T States	4 MHz E.T.
SBC A, r	1	4	1.00
SBC A, n	2	7(4, 3)	1.75
SBC A, (HL)	2	7 (4, 3)	1.75
SBC A, (IX+d)	5	19 (4, 4, 3, 5, 3)	4.75
SBC A, (lY+d)	5	19 (4, 4, 3, 5, 3)	4.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is set if borrow from bit 4; reset otherwise
P/V is reset if overflow; reset otherwise
N is set
C is 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`, at execution of `SBC A, (HL)` the Accumulator contains `10H`.

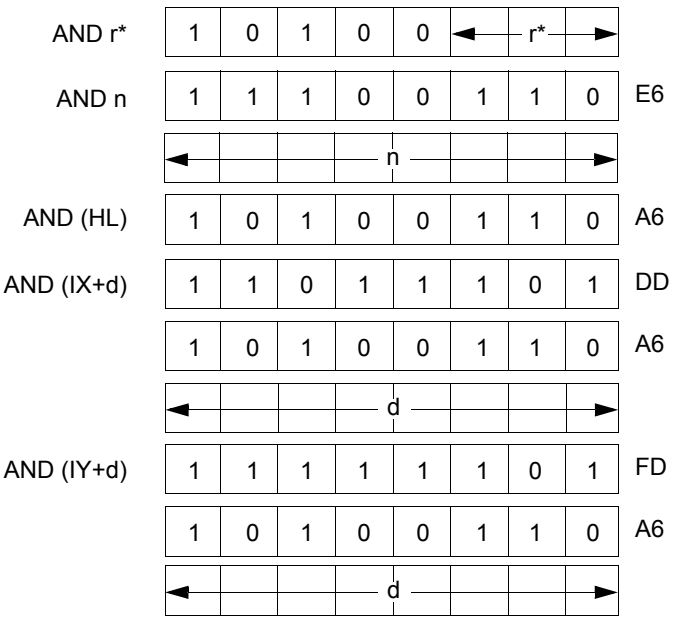
AND s

Operation: $A \leftarrow A \wedge s$

Op Code: AND

Operands: s

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



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

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

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

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

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is set
P/V is reset if overflow; reset otherwise
N is reset
C is reset

Example: If the B register contains 7BH (0111 1011), and the Accumulator contains C3H (1100 0011), at execution of AND B the Accumulator contains 43H (0100 0011).

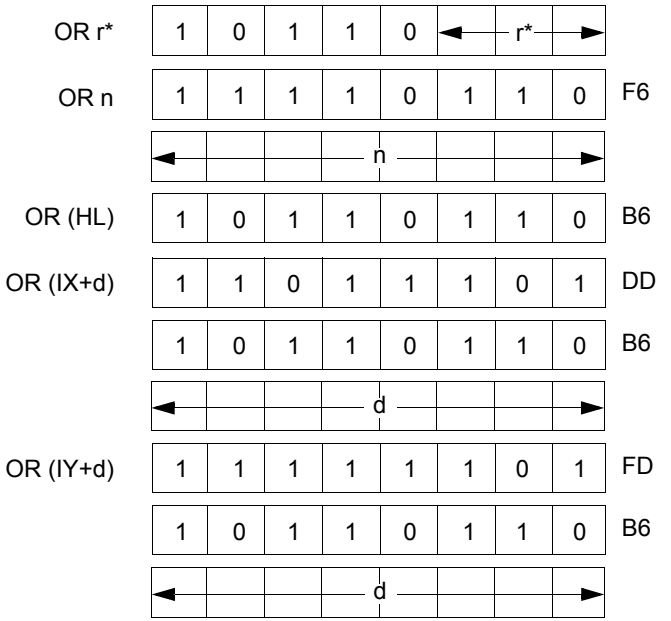
OR s

Operation: $A \leftarrow A \vee s$

Op Code: OR

Operands: s

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



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

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

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

Instruction	M cycles	T States	4 MHz E.T.
OR r	1	4	1.00
OR n	2	7 (4, 3)	1.75
OR (HL)	2	7 (4, 3)	1.75
OR (IX+d)	5	19 (4, 4, 3, 5, 3)	4.75
OR (lY+d)	5	19 (4, 4, 3, 5, 3)	4.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is reset
P/V is set if overflow; reset otherwise
N is reset
C is reset

Example: If the H register contains 48H (0100 0100), and the Accumulator contains 12H (0001 0010), at execution of OR H the Accumulator contains 5AH (0101 1010).

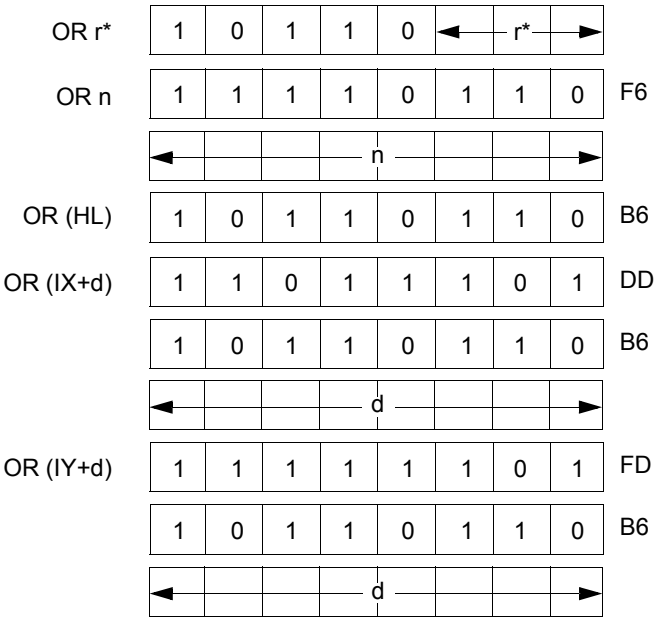
XOR s

Operation: A ← A ⊕ s

Op Code: XOR

Operands: s

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



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

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

Description: The logical exclusive-OR operation is performed between the byte specified by the s operand and the byte contained in the Accumulator; the result is stored in the Accumulator.

Instruction	M Cycles	T States	4 MHz E.T.
XOR r	1	4	1.00
XOR n	2	7 (4, 3)	1.75
XOR (HL)	2	7 (4, 3)	1.75
XOR (IX+d)	5	19 (4, 4, 3, 5, 3)	4.75
XOR (lY+d)	5	19 (4, 4, 3, 5, 3)	4.75

Condition Bits Affected:

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

Example: If the Accumulator contains 96H (1001 0110), at execution of XOR 5DH (5DH = 0101 1101) the Accumulator contains CBH (1100 1011).

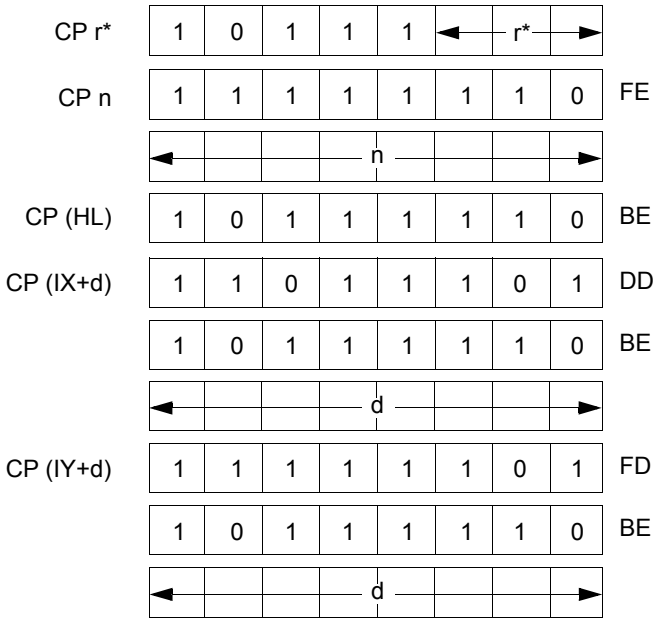
CP s

Operation: A - s

Op Code: CP

Operands: s

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



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

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

Description: The contents of the s operand are compared with the contents of the Accumulator. If there is a true compare, the Z flag is set. The execution of this instruction does not affect the contents of the Accumulator.

Instruction	M Cycles	T States	4 MHz E.T.
CP r	1	4	1.00
CP n	2	7(4, 3)	1.75
CP (HL)	2	7 (4, 3)	1.75
CP (IX+d)	5	19 (4, 4, 3, 5, 3)	4.75
CP (lY+d)	5	19 (4, 4, 3, 5, 3)	4.75

Condition Bits Affected:

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

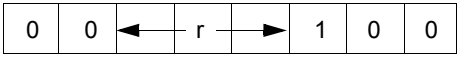
Example: If the Accumulator contains 63H, the HL register pair contains 6000H, and memory location 6000H contains 60H, the instruction `CP (HL)` results in the PN flag in the F register resetting.

INC r

Operation: r ← r + 1

Op Code: INC

Operands: r



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

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

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is set if carry from bit 3; reset otherwise
P/V is set if r was 7FH before operation; reset otherwise
N is reset
C is not affected

Example: If the contents of register D are 28H, at execution of `INC D` the contents of register D are 29H.

INC (HL)

Operation: (HL) ← (HL) + 1

Op Code: INC

Operands: (HL)

0	0	1	1	0	1	0	0	34

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

M Cycles	T States	4 MHz E.T.
3	11 (4, 4, 3)	2.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is set if carry from bit 3; reset otherwise
P/V is set if (HL) was 7FH before operation; reset otherwise
N is reset
C is not affected

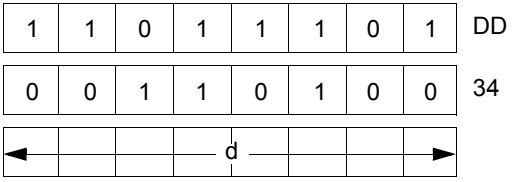
Example: If the contents of the HL register pair are `3434H`, and the contents of address `3434H` are `82H`, at execution of `INC (HL)` memory location `3434H` contains `83H`.

INC (IX+d)

Operation: (IX+d) ← (IX+d) + 1

Op Code: INC

Operands: (IX+d)



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

M Cycles	T States	4 MHz E.T.
6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is set if carry from bit 3; reset otherwise
P/V is set if (IX+d) was 7FH before operation; reset otherwise
N is reset
C is not affected

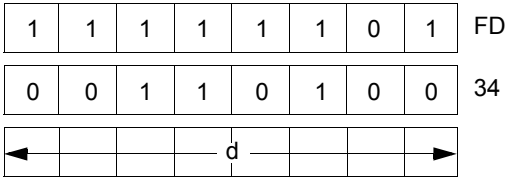
Example: If the contents of the Index Register pair IX are `2020H`, and the memory location `2030H` contains byte `34H`, at execution of `INC (IX+10H)` the contents of memory location `2030H` is `35H`.

INC (IY+d)

Operation: (IY+d) ← (IY+d) + 1

Op Code: INC

Operands: (IY+d)



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

M Cycles	T States	4 MHz E.T.
6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is set if carry from bit 3; reset otherwise
P/V is set if (IY+d) was 7FH before operation; reset otherwise
N is reset
C is not affected

Example: If the contents of the Index Register pair IY are `2020H`, and the memory location `2030H` contain byte `34H`, at execution of `INC (IY+10H)` the contents of memory location `2030H` are `35H`.

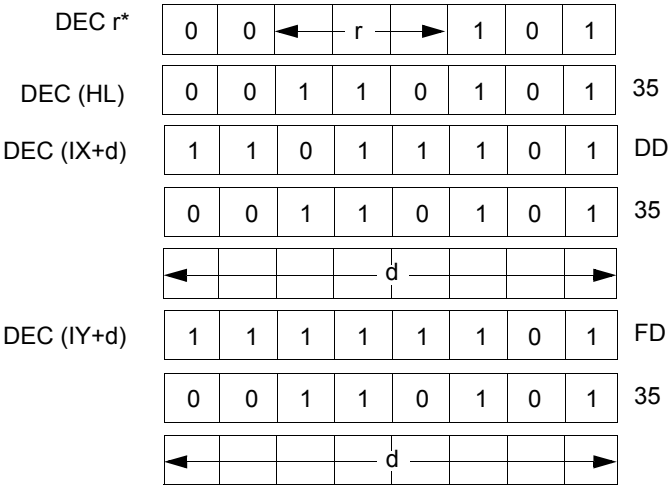
DEC m

Operation: m ← m- 1

Op Code: DEC

Operands: m

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



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

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

Description: The byte specified by the m operand is decremented.

Instruction	M Cycles	T States	4 MHz E.T.
DEC r	1	4	1.00
DEC (HL)	3	11 (4, 4, 3)	2.75
DEC (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75
DEC (lY+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is set if borrow from bit 4, reset otherwise
P/V is set if m was 80H before operation; reset otherwise
N is set
C is not affected

Example: If the D register contains byte 2AH, at execution of `DEC D` register D contains 29H.

General-Purpose Arithmetic and CPU Control Groups

DAA

Operation:

Op Code: DAA

0	0	1	0	0	1	1	1	27

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

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected:

S is set if most-significant bit of Accumulator is 1 after operation; reset otherwise
Z is set if Accumulator is zero after operation; reset otherwise
H, see instruction
P/V is set if Accumulator is even parity after operation; reset otherwise
N is not affected
C, see instruction

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

```
 15
+27
 42
```

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

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

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

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

CPL

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

Op Code: CPL

0	0	1	0	1	1	1	1	2F

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

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected:

S is not affected
Z is not affected
H is set
P/V is not affected
N is set
C is not affected

Example: If the contents of the Accumulator are `1011 0100`, at execution of `CPL` the Accumulator contents are `0100 1011`.

NEG

Operation: A ← 0 - A

Op Code: NEG

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.

M Cycles	T States	4 MHz E.T.
2	8 (4, 4)	2.00

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is 0; reset otherwise
H is set if borrow from bit 4; reset otherwise
P/V is set if Accumulator was `80H` before operation; reset otherwise
N is set
C is set if Accumulator was not `00H` before operation; reset otherwise

Example: If the contents of the Accumulator are

1	0	0	1	1	0	0	0

at execution of `NEG` the Accumulator contents are

0	1	1	0	1	0	0	0

CCF

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

Op Code: CCF

0	0	1	1	1	1	1	1	3F

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

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected:

S is not affected
Z is not affected
H, previous carry is copied
P/V is not affected
N is reset
C is set if CY was 0 before operation; reset otherwise

SCF

Operation: CY ← 1

Op Code: SCF

0	0	1	1	0	1	1	1	37

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

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected:

S is not affected
Z is not affected
H is reset
P/V is not affected
N is reset
C is set

NOP

Operation: —

Op Code: NOP

0	0	0	0	0	0	0	0	00

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

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected: None

HALT

Operation: —

Op Code: HALT

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 executes NOPs to maintain memory refresh logic.

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected: None

DI

Operation: IFF ← 0

Op Code: DI

1	1	1	1	0	0	1	1	F3

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

M cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected: None

Example: When the CPU executes the instruction `DI` the maskable interrupt is disabled until it is subsequently re-enabled by an EI instruction. The CPU does not respond to an Interrupt Request (INT) signal.

EI

Operation: IFF ← 1

Op Code: EI

1	1	1	1	1	0	1	1	FB

Description: The enable interrupt instruction sets both interrupt enable flip flops (IFFI and IFF2) to a logic 1, allowing recognition of any maskable interrupt. Note that during the execution of this instruction and the following instruction, maskable interrupts are disabled.

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected: None

Example: When the CPU executes instruction `El RETI` the maskable interrupt is enabled at execution of the `RETI` instruction.

IM 0

Operation: —

Op Code: IM

Operands: 0

1	1	1	0	1	1	0	1	ED
0	1	0	0	0	1	1	0	46

Description: The IM 0 instruction sets interrupt mode 0. In this mode, the interrupting device can insert any instruction on the data bus for execution by the CPU. The first byte of a multi-byte instruction is read during the interrupt acknowledge cycle. Subsequent bytes are read in by a normal memory read sequence.

M Cycles	T States	4 MHz E.T.
2	8 (4, 4)	2.00

Condition Bits Affected: None

IM 1

Operation: —

Op Code: IM

Operands: 1

1	1	1	0	1	1	0	1	ED
0	1	0	1	0	1	1	0	56

Description: The IM 1 instruction sets interrupt mode 1. In this mode, the processor responds to an interrupt by executing a restart to location 0038H.

M Cycles	T States	4 MHz E.T.
2	8 (4, 4)	2.00

Condition Bits Affected: None

IM 2

Operation: —

Op Code: IM

Operands: 2

1	1	1	0	1	1	0	1	ED
0	1	0	1	1	1	1	0	5E

Description: The IM 2 instruction sets the vectored interrupt mode 2. This mode allows an indirect call to any memory location by an 8-bit vector supplied from the peripheral device. This vector then becomes the least-significant eight bits of the indirect pointer, while the I register in the CPU provides the most-significant eight bits. This address points to an address in a vector table that is the starting address for the interrupt service routine.

M Cycles	T States	4 MHz E.T.
2	8 (4, 4)	2.00

Condition Bits Affected: None

16-Bit Arithmetic Group

ADD HL, ss

Operation: HL ← HL + ss

Op Code: ADD

Operands: HL, ss

0	0	s	s	1	0	0	1

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

Register Pair	ss
BC	00
DE	01
HL	10
SP	11

M Cycles	T States	4 MHz E.T.
3	11 (4, 4, 3)	2.75

Condition Bits Affected:

S is not affected
Z is not affected
H is set if carry out of bit 11; reset otherwise
P/V is not affected
N is reset
C is set if carry from bit 15; reset otherwise

Example: If register pair HL contains the integer `4242H`, and register pair DE contains `1111H`, at execution of `ADD HL, DE` the HL register pair contains `5353H`.

ADC HL, ss

Operation: HL ← HL + ss + CY

Op Code: ADC

Operands: HL, ss

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) are added with the Carry flag (C flag in the F register) to the contents of register pair HL, and the result is stored in HL. Operand ss is specified as follows in the assembled object code.

Register Pair	ss
BC	00
DE	01
HL	10
SP	11

M Cycles	T States	4 MHz E.T.
4	15 (4, 4, 4, 3)	3.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
R is set if carry out of bit 11;. reset otherwise
P/V is set if overflow; reset otherwise
N is reset
C is 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, at execution of `ADC HL, BC` the contents of HL are `765AH`.

SBC HL, ss

Operation: HL ← HI - ss - CY

Op Code: SBC

Operands: HL, ss

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 (C flag in the F register) are subtracted from the contents of register pair HL, and the result is stored in HL. Operand ss is specified as follows in the assembled object code.

Register Pair	ss
BC	00
DE	01
HL	10
SP	11

M Cycles	T States	4 MHz E.T.
4	15 (4, 4, 4, 3)	3.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is set if a borrow from bit 12; reset otherwise
P/V is set if overflow; reset otherwise
N is set
C is 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. At execution of `SBC HL, DE` the contents of HL are `8887H`.

ADD IX, pp

Operation: IX ← IX + pp

Op Code: ADD

Operands: IX, pp

1	1	0	1	1	1	0	1	DD
0	0	p	p	1	0	0	1	

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

Register Pair	pp
BC	00
DE	01
IX	10
SP	11

M Cycles	T States	4 MHz E.T.
4	15 (4, 4, 4, 3)	3.75

Condition Bits Affected:

S is not affected
Z is not affected
H is set if carry out of bit 11; reset otherwise
P/V is not affected
N is reset
C is set if carry from bit 15; reset otherwise

Example: If the contents of Index Register IX are 333H, and the contents of register pair BC are 5555H, at execution of `ADD IX, BC` the contents of IX are 8888H.

ADD IY, rr

Operation: IY ← IY + rr

Op Code: ADD

Operands: IY, rr

1	1	1	1	1	1	0	1	FD
0	0	r	r	1	0	0	1	

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

Register Pair	rr
BC	00
DE	01
IY	10
SP	11

M Cycles	T States	4 MHz E.T.
4	15 (4, 4, 4, 3)	3.75

Condition Bits Affected:

S is not affected
Z is not affected
H is set if carry out of bit 11; reset otherwise
P/V is not affected
N is reset
C is set if carry from bit 15; reset otherwise

Example: If the contents of Index Register IY are 333H, and the contents of register pair BC are 555H, at execution of `ADD IY, BC` the contents of IY are 8888H.

INC ss

Operation: ss ← ss + 1

Op Code: INC

Operands: ss

0	0	s	s	0	0	1	1

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

Register Pair	ss
BC	00
DE	01
HL	10
SP	11

M Cycles	T States	4 MHz E.T.
1	6	1.50

Condition Bits Affected: None

Example: If the register pair contains 1000H, after the execution of `INC HL, HL` contains 1001H.

INC IX

Operation: IX ← IX + 1

Op Code: INC

Operands: IX

1	1	0	1	1	1	0	1	DD
0	0	1	0	0	0	1	1	23

Description: The contents of the Index Register IX are incremented.

M Cycles	T States	4 MHz E.T.
2	10 (4, 6)	2.50

Condition Bits Affected: None

Example: If the Index Register IX contains the integer `3300H`. at execution of `INC IX` the contents of Index Register IX are `3301H`.

INC IY

Operation: IY ← IY + 1

Op Code: INC

Operands: IY

1	1	1	1	1	1	0	1	FD
0	0	1	0	0	0	1	1	23

Description: The contents of the Index Register IY are incremented.

M Cycles	T States	4 MHz E.T.
2	10 (4, 6)	2.50

Condition Bits Affected: None

Example: If the contents of the Index Register are `2977H`, at execution of `INC IY` the contents of Index Register IY are `2978H`.

DEC ss

Operation: ss ← ss - 1

Op Code: DEC

Operands: ss

0	0	s	s	1	0	1	1

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

Register Pair	ss
BC	00
DE	01
HL	10
SP	11

M Cycles	T States	4 MHz E.T.
1	6	1.50

Condition Bits Affected: None

Example: If register pair HL contains `1001H`, at execution of `DEC HL` the contents of HL are `1000H`.

DEC IX

Operation: IX ← IX - 1

Op Code: DEC

Operands: IX

1	1	0	1	1	1	0	1	DD
0	0	1	0	1	0	1	1	2B

Description: The contents of Index Register IX are decremented.

M Cycles	T States	4 MHz E.T.
2	10 (4, 6)	2.50

Condition Bits Affected: None

Example: If the contents of Index Register IX are `2006H`, at execution of `DEC IX` the contents of Index Register IX are `2005H`.

DEC IY

Operation: IY ← IY - 1

Op Code: DEC

Operands: IY

1	1	1	1	1	1	0	1	FD
0	0	1	0	1	0	1	1	2B

Description: The contents of the Index Register IY are decremented.

M Cycles	T States	4 MHz E.T.
2	10 (4, 6)	2.50

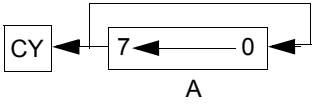
Condition Bits Affected: None

Example: If the contents of the index Register IY are `7649H`, at execution of `DEC IY` the contents of index Register IY are `7648H`.

Rotate and Shift Group

RLCA

Operation:



Op Code: RLCA

Operands: —

0	0	0	0	0	1	1	1	07

Description: The contents of the Accumulator (register A) are rotated left 1-bit position. The sign bit (bit 7) is copied to the Carry flag and also to bit 0. Bit 0 is the least-significant bit.

M cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected:

S is not affected
Z is not affected
H is reset
P/V is not affected
N is reset
C is data from bit 7 of Accumulator

Example: If the contents of the Accumulator are

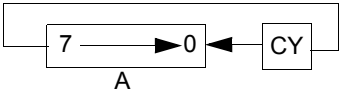
7	6	5	4	3	2	1	0
1	0	0	0	1	0	0	0

at execution of `RLCA` the contents of the Accumulator and Carry flag are

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

RLA

Operation:



Op Code: RLA

Operands: —

0	0	0	1	0	1	1	1	17

Description: The contents of the Accumulator (register A) are rotated left 1-bit position through the Carry flag. The previous content of the Carry flag is copied to bit 0. Bit 0 is the least-significant bit.

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected: Condition Bits Affected

S is not affected
Z is not affected
H is reset
P/V is not affected
N is reset
C is data from bit 7 of Accumulator

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

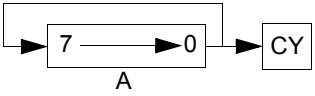
C	7	6	5	4	3	2	1	0
1	0	1	1	1	0	1	1	0

at execution of `RLA` the contents of the Accumulator and the Carry flag are

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

RRCA

Operation:



Op Code: RRCA

Operands: —

0	0	0	0	1	1	1	1	0F

Description: The contents of the Accumulator (register A) are rotated right 1-bit position. Bit 0 is copied to the Carry flag and also to bit 7. Bit 0 is the least-significant bit.

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected:

S is not affected
Z is not affected
H is reset
P/V is not affected
N is reset
C is data from bit 0 of Accumulator

Example: If the contents of the Accumulator are

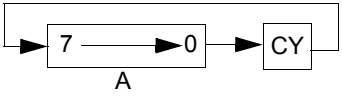
7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	1

at execution of `RRCA` the contents of the Accumulator and the Carry flag are

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

RRA

Operation:



Op Code: RRA

Operands: —

0	0	0	1	1	1	1	1	1F

Description: The contents of the Accumulator (register A) are rotated right 1-bit position through the Carry flag. The previous content of the Carry flag is copied to bit 7. Bit 0 is the least-significant bit.

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected:

S is not affected
Z is not affected
H is reset
P/V is not affected
N is reset
C is data from bit 0 of Accumulator

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

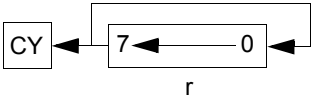
7	6	5	4	3	2	1	0	C
1	1	1	0	0	0	0	1	0

at execution of `RRA` the contents of the Accumulator and the Carry flag are

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

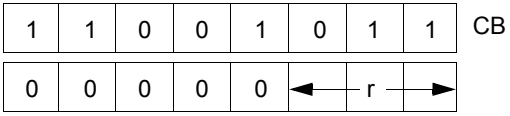
RLC r

Operation:



Op Code: RLC

Operands: r



Description: The contents of register r are rotated left 1-bit position. The content of bit 7 is copied to the Carry flag and also to bit 0. Operand r is specified as follows in the assembled object code:

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

M Cycles	T States	4 MHz E.T.
2	8 (4, 4)	2.00

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is reset
P/V is set if parity even; reset otherwise
N is reset
C is data from bit 7 of source register

Example: If the contents of register r are

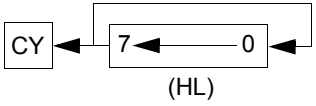
7	6	5	4	3	2	1	0
1	0	0	0	1	0	0	0

at execution of `RLC r` the contents of register r and the Carry flag are

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

RLC (HL)

Operation:



Op Code: RLC

Operands: (HL)

1	1	0	0	1	0	1	1	CB
0	0	0	0	0	1	1	0	06

Description: The contents of the memory address specified by the contents of register pair HL are rotated left 1-bit position. The content of bit 7 is copied to the Carry flag and also to bit 0. Bit 0 is the least-significant bit.

M Cycles	T States	4 MHz E.T.
4	15 (4, 4, 4, 3)	3.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is reset
P/V is set if parity even; reset otherwise
N is reset
C is data from bit 7 of source register

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

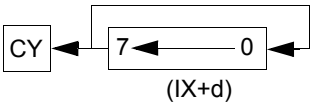
7	6	5	4	3	2	1	0
1	0	0	0	1	0	0	0

at execution of `RLC(HL)` the contents of memory location 2828H and the Carry flag are

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

RLC (IX+d)

Operation:



Op Code: RLC

Operands: (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

Description: The contents of the memory address specified by the sum of the contents of the Index Register IX and a two's complement displacement integer d, are rotated left 1-bit position. The content of bit 7 is copied to the Carry flag and also to bit 0. Bit 0 is the least-significant bit.

M Cycles	T States	4 MHz E.T.
6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is reset
P/V is set if parity even; reset otherwise
N is reset
C is data from bit 7 of source register

Example: If the contents of the Index Register IX are `1000H`, and the contents of memory location `1022H` are

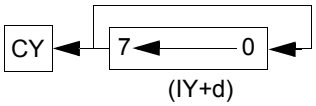
7	6	5	4	3	2	1	0
1	0	0	0	1	0	0	0

at execution of `RLC (IX+2H)` the contents of memory location `1002H` and the Carry flag are

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

RLC (IY+d)

Operation:



Op Code: RLC

Operands: (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 memory address specified by the sum of the contents of the Index Register IY and a two's complement displacement integer d are rotated left 1-bit position. The content of bit 7 is copied to the Carry flag and also to bit 0. Bit 0 is the least-significant bit.

M Cycles	T States	4 MHz E.T.
6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is reset
P/V is set if parity even; reset otherwise
N is reset
C is data from bit 7 of source register

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

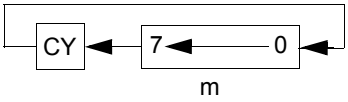
7	6	5	4	3	2	1	0
1	0	0	0	1	0	0	0

at execution of `RLC (IY+2H)` the contents of memory location `1002H` and the Carry flag are

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

RL m

Operation:



Op Code: PL

Operands: m

The m operand is any of r, (HL), (IX+d), or (lY+d), as defined for the analogous PLC instructions. These possible Op Code/operand combinations are specified as follows in the assembled object code:

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	FB
	1	1	0	0	1	0	1	1	CB
	d								
	0	0	0	1	0	1	1	0	16

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

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

Description: The contents of the m operand are rotated left 1-bit position. The content of bit 7 is copied to the Carry flag and the previous content of the Carry flag is copied to bit 0.

Instruction	M Cycles	T States	4 MHz E.T.
RL r	2	8 (4, 4)	2.00
RL (HL)	4	15(4, 4, 4, 3)	3.75
RL (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75
RL (IY+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is reset
P/V is set if parity even; reset otherwise
N is reset
C is data from bit 7 of source register

Example: If the contents of register D and the Carry flag are

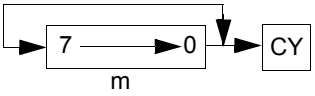
C	7	6	5	4	3	2	1	0
0	1	0	0	0	1	1	1	1

at execution of `RL D` the contents of register D and the Carry flag are

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

RRC m

Operation:



Op Code: RRC

Operands: m

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

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	OE
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	OE
RRC (IY+d)	1	1	1	1	1	1	0	1	FB
	1	1	0	0	1	0	1	1	CB
	←			d				→	
	0	0	0	0	1	1	1	0	OE

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

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

Description: The contents of the m operand are rotated right 1-bit position. The content of bit 0 is copied to the Carry flag and also to bit 7. Bit 0 is the least-significant bit.

Instruction	M cycles	T States	4 MHz E.T.
RRC r	2	8 (4, 4)	2.00
RRC (HL)	4	15 (4, 4, 4, 3)	3.75
RRC (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75
RRC (lY+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is reset
P/V is set if parity even; reset otherwise,
N is reset
C is data from bit 0 of source register

Example: If the contents of register A are

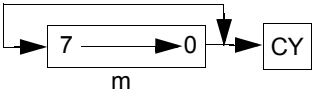
7	6	5	4	3	2	1	0
0	0	1	1	0	0	0	1

at execution of `RRC A` the contents of register A and the Carry flag are

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

RR m

Operation:



Op Code: RR

Operands: m

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

RR r*	1	1	0	0	1	0	1	1	CB
	0	0	0	0	1	r*			
RR (HL)	1	1	0	0	1	0	1	1	CB
	0	0	0	0	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

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

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

Description: The contents of operand m are rotated right 1-bit position through the Carry flag. The content of bit 0 is copied to the Carry flag and the previous content of the Carry flag is copied to bit 7. Bit 0 is the least-significant bit.

Instruction	M Cycles	T States	4 MHz E.T.
RR r	2	8 (4, 4)	2.00
RR (HL)	4	15 (4, 4, 4, 3)	3.75
RR (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75
RR (lY+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is reset
P/V is set if parity even; reset otherwise,
N is reset
C is data from bit 0 of source register

Example: If the contents of the HL register pair are 4343H, and the contents of memory location 4343H and the Carry flag are

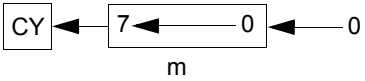
7	6	5	4	3	2	1	0	C
1	1	0	1	1	1	0	1	0

at execution of `RR (HL)` the contents of location 4343H and the Carry flag are

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

SLA m

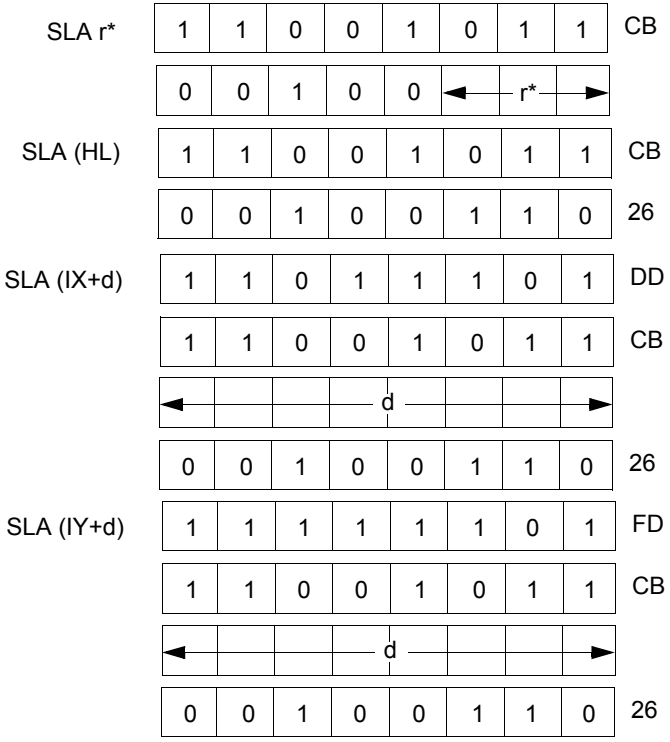
Operation:



Op Code: SLA

Operands: m

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



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

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

Description: An arithmetic shift left 1-bit position is performed on the contents of operand m. The content of bit 7 is copied to the Carry flag. Bit 0 is the least-significant bit.

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

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is reset
P/V is set if parity is even; reset otherwise
N is reset
C is data from bit 7

Example: If the contents of register L are

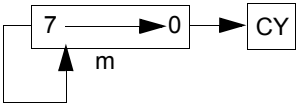
7	6	5	4	3	2	1	0
1	0	1	1	0	0	0	1

at execution of `SLA L` the contents of register L and the Carry flag are

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

SRA m

Operation:



Op Code: SRA

Operands: m

The m operand is any of r, (HL), (IX+d), or (lY+d), as defined for the analogous PLC instructions. These possible Op Code/operand combinations are specified as follows in the assembled object code:

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

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

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

Description: An arithmetic shift right 1-bit position is performed on the contents of operand m. The content of bit 0 is copied to the Carry flag and the previous content of bit 7 is unchanged. Bit 0 is the least-significant bit.

Instruction	M Cycles	T States	4 MHz E.T.
SRA r	2	8 (4, 4)	2.00
SRA (HL)	4	15 (4, 4, 4, 3)	3.75
SRA (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75
SRA (lY+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected:

S is set if result is negative; reset otherwise
Z is set if result is zero; reset otherwise
H is reset
P/V is set if parity is even; reset otherwise
N is reset
C is data from bit 0 of source register

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

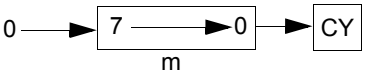
7	6	5	4	3	2	1	0
1	0	1	1	1	0	0	0

at execution of `SRA (IX+3H)` the contents of memory location 1003H and the Carry flag are

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

SRL m

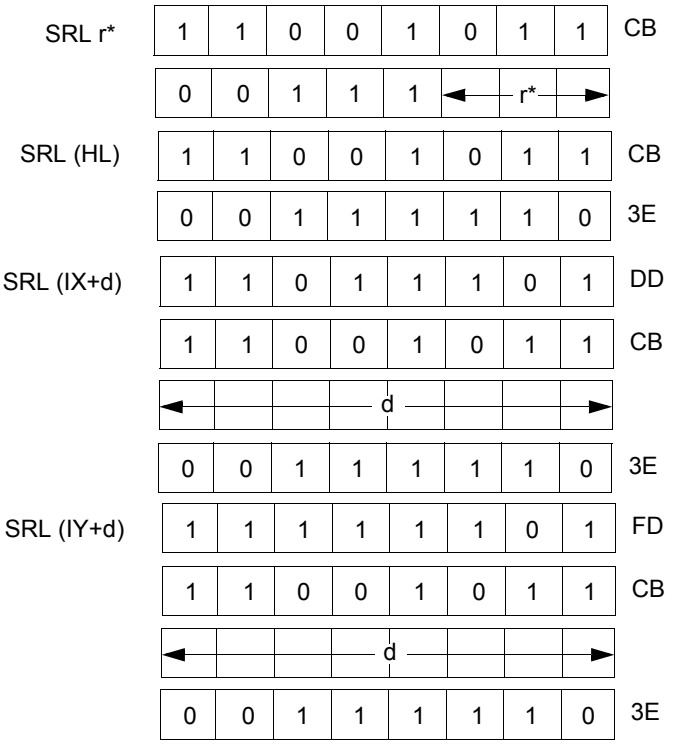
Operation:



Op Code: SRL

Operands: m

The operand m is any of r, (HL), (IX+d), or (lY+d), as defined for the analogous RLC instructions. These possible Op Code/operand combinations are specified as follows in the assembled object code



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

Description: The contents of operand m are shifted right 1-bit position. The content of bit 0 is copied to the Carry flag, and bit 7 is reset. Bit 0 is the least-significant bit.

Instruction	M Cycles	T States	4 MHz E.T.
SRL r	2	8 (4, 4)	2.00
SRL (HL)	4	15 (4, 4, 4, 3)	3.75
SRL (1X+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75

SRL (lY+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected:

S is reset
Z is set if result is zero; reset otherwise
H is reset
P/V is set if parity is even; reset otherwise
N is reset
C is data from bit 0 of source register

Example: If the contents of register B are

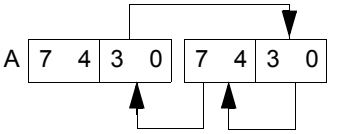
7	6	5	4	3	2	1	0
1	0	0	0	1	1	1	1

at execution of `SRL B` the contents of register B and the Carry flag are

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

RLD

Operation:



Op Code: RLD

Operands: —

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 to 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 to 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 to the low order four bits of memory location (HL). The contents of the high order bits of the Accumulator are unaffected.

Note: (HL) means the memory location specified by the contents of the HL register pair.

M Cycles	T States	4 MHz E.T.
5	18 (4, 4, 3, 4, 3)	4.50

Condition Bits Affected:

S is set if Accumulator is negative after operation; reset otherwise
Z is set if Accumulator is zero after operation; reset otherwise
H is reset
P/V is set if parity of Accumulator is even after operation; reset otherwise
N is reset
C is 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

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

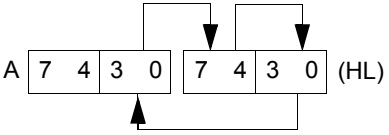
at execution of `RLD` the contents of the Accumulator and memory location `5000H` are

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

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

RRD

Operation:



Op Code: RRD

Operands: —

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 to the low order four bits of the Accumulator (register A). The previous contents of the low order four bits of the Accumulator are copied to 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 to the low order four bits of (HL). The contents of the high order bits of the Accumulator are unaffected.

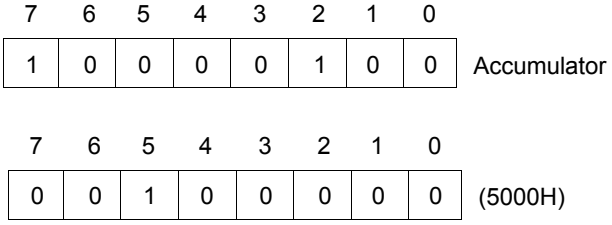
(HL) means the memory location specified by the contents of the HL register pair.

M Cycles	T States	4 MHz E.T.
5	18 (4, 4, 3, 4, 3)	4.50

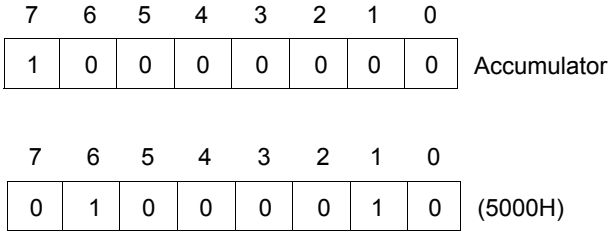
Condition Bits Affected:

S is set if Accumulator is negative after operation; reset otherwise
Z is set if Accumulator is zero after operation; reset otherwise
H is reset
P/V is set if parity of Accumulator is even after operation; reset otherwise
N is reset
C is not affected

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



at execution of `RRD` the contents of the Accumulator and memory location `5000H` are



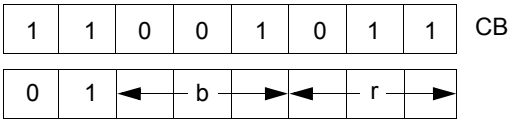
Bit Set, Reset, and Test Group

BIT b, r

Operation: $Z \leftarrow \overline{rb}$

Op Code: BIT

Operands: b, r



Description: This instruction tests bit b in register r and sets the Z flag accordingly. Operands b and r are specified as follows in the assembled object code:

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

M Cycles	T States	4 MHz E.T.
2	8 (4, 4)	4.50

Condition Bits Affected:

S is unknown
Z is set if specified bit is 0; reset otherwise
H is set
P/V is unknown
N is reset
C is not affected

Example: If bit 2 in register B contains 0, at execution of `BIT 2, B` the Z flag in the F register contains 1, and bit 2 in register B remains 0. Bit 0 in register B is the least-significant bit.

BIT b, (HL)

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

Op Code: BIT

Operands: b, (HL)

1	1	0	0	1	0	1	1	CB
0	1		b		1	1	0	

Description: This instruction tests bit b in the memory location specified by the contents of the HL register pair and sets the Z flag accordingly. Operand b is specified as follows in the assembled object code:

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

M Cycles	T States	4 MHz E.T.
3	12 (4, 4, 4) 4	3.00

Condition Bits Affected:

S is unknown
Z is set if specified Bit is 0; reset otherwise
H is set
P/V is unknown
H is reset
C is not affected

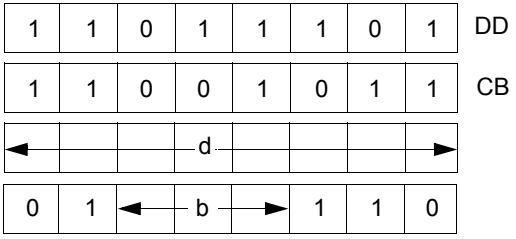
Example: If the HL register pair contains `4444H`, and bit 4 in the memory location `444H` contains 1, at execution of `BIT 4, (HL)` the Z flag in the F register contains 0, and bit 4 in memory location `4444H` still contains 1. Bit 0 in memory location `4444H` is the least-significant bit.

BIT b, (IX+d)

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

Op Code: BIT

Operands: b, (IX+d)



Description: This instruction tests bit b in the memory location specified by the contents of register pair IX combined with the two's complement displacement d and sets the Z flag accordingly. Operand b is specified as follows in the assembled object code.

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

M cycles	T States	4 MHz E.T.
5	20 (4, 4, 3, 5, 4)	5.00

Condition Bits Affected:

S is unknown
Z is set if specified Bit is 0; reset otherwise
H is set
P/V is unknown
N is reset
C is not affected

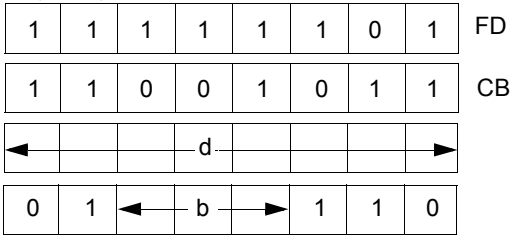
Example: If the contents of Index Register IX are `2000H`, and bit 6 in memory location `2004H` contains 1, at execution of `BIT 6, (IX+4H)` the Z flag in the F register contains 0, and bit 6 in memory location `2004H` still contains 1. Bit 0 in memory location `2004H` is the least-significant bit.

BIT b, (IY+d)

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

Op Code: BIT

Operands: b, (IY+d)



Description: This instruction tests bit b in the memory location specified by the content of register pair IY combined with the two's complement displacement d and sets the Z flag accordingly. Operand b is specified as follows in the assembled object code.

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

M Cycles	T States	4 MHz E.T.
5	20 (4, 4, 3, 5, 4)	5.00

Condition Bits Affected:

S is unknown
Z is set if specified Bit is 0; reset otherwise
H is set
P/V is unknown
H is reset
C is not affected

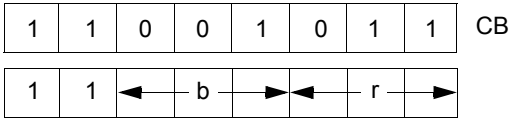
Example: If the contents of Index Register are 2000H, and bit 6 in memory location 2004H contains 1, at execution of BIT 6, (IY+4H) the Z flag and the F register still contain 0, and bit 6 in memory location 2004H still contains 1. Bit 0 in memory location 2004H is the least-significant bit.

SET b, r

Operation: rb ← 1

Op Code: SET

Operands: b, r



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

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

M Cycles	T States	4 MHz E.T.
2	8 (4, 4)	2.00

Condition Bits Affected: None

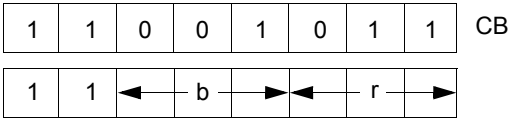
Example: At execution of `SET 4, A` bit 4 in register A sets. Bit 0 is the least-significant bit.

SET b, (HL)

Operation: (HL)b ← 1

Op Code: SET

Operands: b, (HL)



Description: Bit b in the memory location addressed by the contents of register pair HL is set. Operand b is specified as follows in the assembled object code:

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

M Cycles	T States	4 MHz E.T.
4	15 (4, 4, 4, 3)	3.75

Condition Bits Affected: None

Example: If the contents of the HL register pair are `3000H`, at execution of `SET 4, (HL)` bit 4 in memory location `3000H` is 1. Bit 0 in memory location `3000H` is the least-significant bit.

SET b, (IX+d)

Operation: (IX+d)b ← 1

Op Code: SET

Operands: b, (IX+d)

Description: Bit b in the memory location addressed by the sum of the contents of the IX register pair and the two's complement integer d is set. Operand b is specified as follows in the assembled object code:

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

M Cycles	T States	4 MHz E.T.
6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected: None

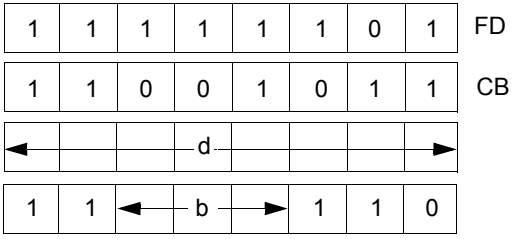
Example: If the contents of Index Register are 2000H, at execution of SET 0, (IX + 3H) bit 0 in memory location 2003H is 1. Bit 0 in memory location 2003H is the least-significant bit.

SET b, (IY+d)

Operation: (IY + d) b ← 1

Op Code: SET

Operands: b, (IY + d)



Description: Bit b in the memory location addressed by the sum of the contents of the IY register pair and the two's complement displacement d is set. Operand b is specified as follows in the assembled object code:

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

M Cycles	T States	4 MHz E.T.
6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected: None

Example: If the contents of Index Register IY are `2000H`, at execution of `SET 0, (IY+3H)` bit 0 in memory location `2003H` is 1. Bit 0 in memory location `2003H` is the least-significant bit.

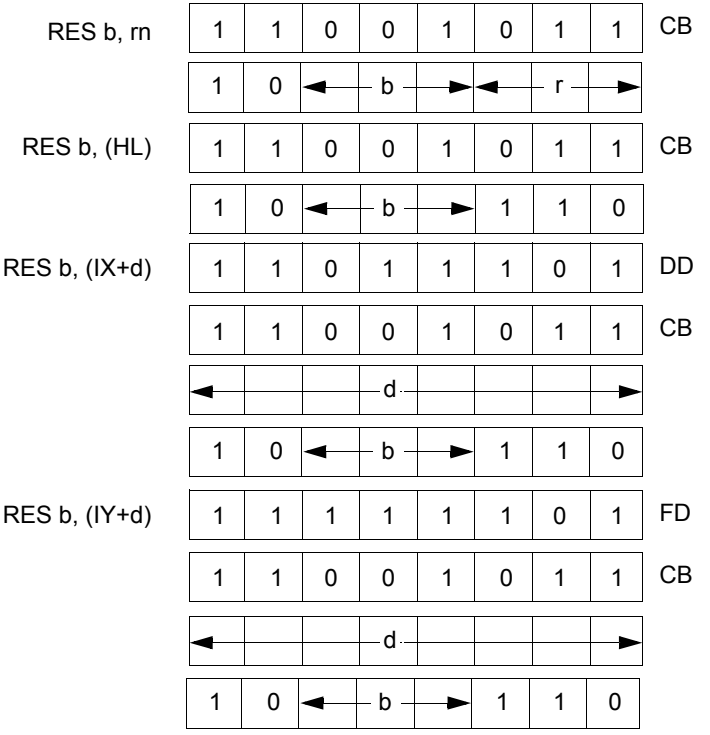
RES b, m

Operation: sb ← 0

Op Code: RES

Operands: b, m

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



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

Description: Bit b in operand m is reset.

Instruction	M Cycles	T States	4 MHz E.T.
RES r	4	8 (4, 4)	2.00
RES (HL)	4	15 (4, 4, 4, 3)	3.75
RES (IX+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75
RES (lY+d)	6	23 (4, 4, 3, 5, 4, 3)	5.75

Condition Bits Affected: None

Example: At execution of `RES 6, D`, bit 6 in register 0 resets. Bit 0 in register D is the least-significant bit.

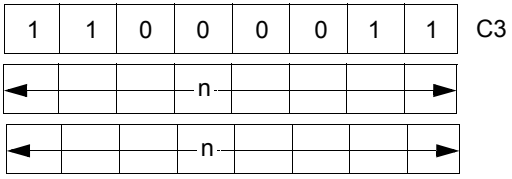
Jump Group

JP nn

Operation: PC ← nn

Op Code: JP

Operands: nn



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

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

M Cycles	T States	4 MHz E.T.
3	10 (4, 3, 3)	2.50

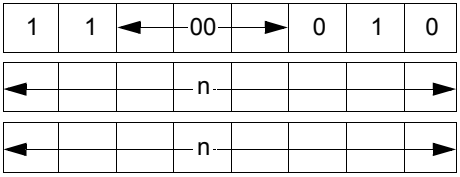
Condition Bits Affected: None

JP cc, nn

Operation: IF cc true, PC ← nn

Op Code: JP

Operands: cc, nn



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

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

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

M Cycles	T States	4 MHz E.T.
3	10 (4, 3, 3)	2.50

Condition Bits Affected: None

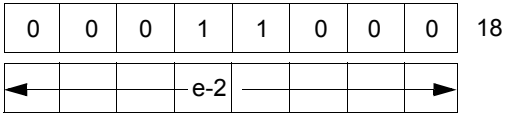
Example: If the Carry flag (C flag in the F register) is set and the contents of address 1520 are `03H`, at execution of `JP C, 1520H` the Program Counter contains `1520H`, and on the next machine cycle the CPD fetches byte `03H` from address `1520H`.

JR e

Operation: PC ← PC + e

Op Code: JR

Operands: e



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

M Cycles	T States	4 MHz E.T.
3	12 (4, 3, 5)	3.00

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

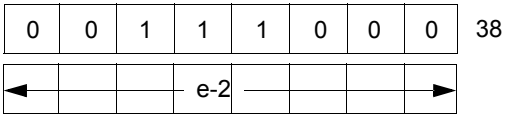
JR C, e

Operation: If C = 0, continue

If C = 1, PC ← PC+ e

Op Code: JR

Operands: C, e



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

If the flag is equal to a 0, the next instruction executed is taken from the location following this instruction. If condition is met

M Cycles	T States	4 MHz E.T.
3	12 (4, 3, 5)	3.00

If condition is not met:

M Cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected: None

Example: The Carry flag is set and it is required to jump back four locations from 480. The assembly language statement is `JR C, $ - 4`

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

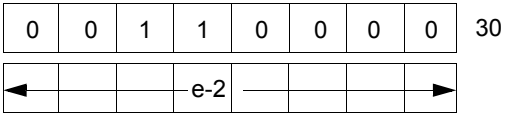
Location	Instruction
47C	← PC after jump
47D	-
47E	-
47F	-
480	38
481	FA (two's complement - 6)

JR NC, e

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

Op Code: JR

Operands: NC, e



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

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

If the condition is met:

M Cycles	T States	4 MHz E.T.
3	12 (4, 3, 5)	3.00

If the condition is not met:

M Cycles	T States	4 MHz E.T.
7	7 (4, 3)	1.75

Condition Bits Affected: None

Example: The Carry Flag is reset and it is required to repeat the jump instruction. The assembly language statement is `JR NC, $`

The resulting object code and PC after the jump are:

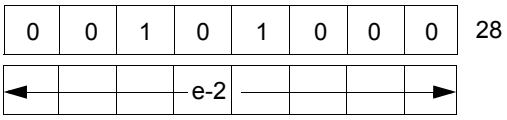
Location	Instruction
480	30 ← PC after jump
481	00

JR Z, e

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

Op Code: JR

Operands: Z, e



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

If the Zero Flag is equal to a 0, the next instruction executed is taken from the location following this instruction. If the condition is met:

M Cycles	T States	4 MHz E.T.
3	12 (4, 3, 5)	3.00

If the condition is not met;

M Cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected: None

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

```
JR Z ,$ + 5
```

The resulting object code and final PC value is:

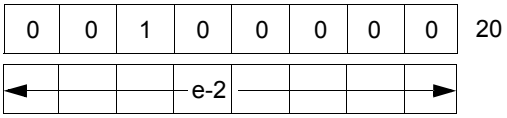
Location	Instruction
300	28
301	03
302	-
303	-
304	-
305	← PC after jump

JR NZ, e

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

Op Code: JR

Operands: NZ, e



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

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

If the condition is met:

M Cycles	T States	4 MHz E.T.
3	12 (4, 3, 5)	3.00

If the condition is not met:

M Cycles	T States	4 MHz E.T.
2	7 (4, 3)	1.75

Condition Bits Affected: None

Example: The Zero Flag is reset and it is required to jump back four locations from 480. The assembly language statement is `JR NZ, $ - 4`

The resulting object code and final PC value is:

Location	Instruction
47C	← PC after jump
47D	-
47E	-
47F	-
480	20
481	FA (two's complement - 6)

JP (HL)

Operation: pc ← hL

Op Code: JP

Operands: (HL)

1	1	1	0	1	0	0	1	E9

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

M Cycles	T States	4 MHz E.T.
1	4	1.00

Condition Bits Affected: None

Example: If the contents of the Program Counter are `1000H`, and the contents of the HL register pair are `4800H`, at execution of `JP (HL)` the contents of the Program Counter are `4800H`.

JP (IX)

Operation: pc ← IX

Op Code: JP

Operands: (IX)

1	1	0	1	1	1	0	1	DD
1	1	1	0	1	0	0	1	E9

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

M Cycles	T States	4 MHz E.T.
2	8 (4, 4)	2.00

Condition Bits Affected: None

Example: If the contents of the Program Counter are `1000H`, and the contents of the IX Register Pair are `4800H`, at execution of `JP (IX)` the contents of the Program Counter are `4800H`.

JP (IY)

Operation: PC ← IY

Op Code: JP

Operands: (IY)

1	1	1	1	1	1	0	1	FD
1	1	1	0	1	0	0	1	E9

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

M Cycles	T States	4 MHz E.T.
2	8 (4, 4)	2.00

Condition Bits Affected: None

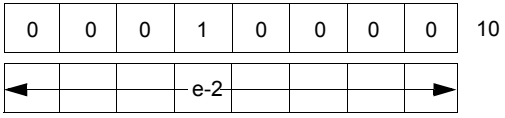
Example: If the contents of the Program Counter are 1000H, and the contents of the IY Register Pair are 4800H, at execution of `JP (IY)` the contents of the Program Counter are 4800H.

DJNZ, e

Operation: -

Op Code: DJNZ

Operands: e



Description: This instruction is similar to the conditional jump instructions except that a register value is used to determine branching. The B register is decremented, and if a non zero value remains, the value of the displacement e is added to the Program Counter (PC). The next instruction is fetched from the location designated by the new contents of the PC. The jump is measured from the address of the instruction Op Code and has a range of -126 to +129 bytes. The assembler automatically adjusts for the twice incremented PC.

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

if B ≠ 0:

M Cycles	T States	4 MHz E.T.
3	13 (5,3, 5)	3.25

If B = 0:

M Cycles	T States	4 MHz E.T.
2	8 (5, 3)	2.00

Condition Bits Affected: None

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

```
          LD   8, 80        ;Set up counter
          LD   HL, Inbuf    ;Set up pointers
          LD   DE, Outbuf

LOOP:     LID  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:
```

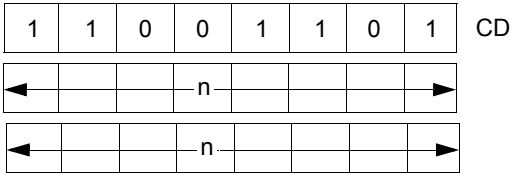
Call And Return Group

CALL nn

Operation: (SP-1) ← PCH, (SP-2) ← PCL, PC ← nn

Op Code: CALL

Operands: nn



The first of the two n operands in the assembled object code above is the least-significant byte of a 2-byte memory address.

Description: The current contents of the Program Counter (PC) are pushed onto the top of the external memory stack. The operands `nn` are then loaded to the PC to point to the address in memory where the first Op Code 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 to the PC. The push is accomplished by first decrementing the current contents of the Stack Pointer (register pair SP), loading the high-order byte of the PC contents to the memory address now pointed to by the SP; then decrementing SP again, and loading the low order byte of the PC contents to the top of stack.

Because this is a 3-byte instruction, the Program Counter was incremented by three before the push is executed.

M Cycles	T States	4 MHz E.T.
5	17 (4, 3, 4, 3, 3)	4.25

Condition Bits Affected: None

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

1A47H contains CDH
IA48H contains 35H
1A49H contains 21H

If an instruction fetch sequence begins, the 3-byte instruction CD3521H is fetched to the CPU for execution. The mnemonic equivalent of this is CALL 2135H. At execution of this instruction, the contents of memory address 3001H is 1AH, the contents of address 3000H is 4AH, the contents of the Stack Pointer is 3000H, and the contents of the Program Counter is 2135H, pointing to the address of the first Op Code of the subroutine now to be executed.

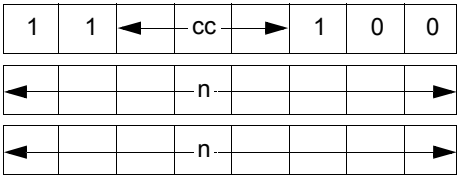
CALL cc, nn

Operation: IF cc true: (sp-1) ← PCH

(sp-2) ← PCL, pc ← nn

Op Code: CALL

Operands: cc, nn



Note: The first of the two n operands in the assembled object code above is the least-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 external memory stack, then loads the operands nn to PC to point to the address in memory where the first Op Code 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 to 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 to the memory address now pointed to by SP; then decrementing SP again, and loading the low order byte of the PC contents to the top of the stack.

Because this is a 3-byte instruction, the Program Counter was incremented by three before the push is executed.

Condition cc is programmed as one of eight status that corresponds to condition bits in the Flag Register (register F). These eight status are

defined in the table below, which also specifies the corresponding cc bit fields in the assembled object code:

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

If cc is true:

M Cycles	T States	4 MHz E.T.
5	17 (4, 3, 4, 3, 3)	4.25

If cc is false:

M Cycles	T States	4 MHz E.T.
3	10 (4, 3, 3)	2.50

Condition Bits Affected: None

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

Location	Contents
1A47H	D4H
1448H	35H
1A49H	21H

then if an instruction fetch sequence begins, the 3-byte instruction D43521H is fetched to the CPU for execution. The mnemonic equivalent of this is CALL NC, 2135H. At execution of this instruction, the contents of memory address 3001H is 1AH, the contents of address 3000H is 4AH, the

contents of the Stack Pointer is 3000H, and the contents of the Program Counter is 2135H, pointing to the address of the first Op Code of the subroutine now to be executed.

RET

Operation: pCL ← (sp), pCH ← (sp+1)

Op Code: RET

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 is 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 is fetched from the memory location specified by the PC. This instruction is normally used to return to the main line program at the completion of a routine entered by a CALL instruction.

M Cycles	T States	4 MHz E.T.
3	10 (4, 3, 3)	2.50

Condition Bits Affected: None

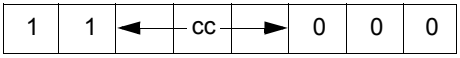
Example: If the contents of the Program Counter are `3535H`, the contents of the Stack Pointer are `2000H`, the contents of memory location `2000H` are `B5H`, and the contents of memory location of memory location `2001H` are `18H`. At execution of `RET` the contents of the Stack Pointer is `2002H`, and the contents of the Program Counter is `18B5H`, pointing to the address of the next program Op Code to be fetched.

RET cc

Operation: If cc true: PCL ← (sp), pCH ← (sp+1)

Op Code: RET

Operands: cc



Description: If condition cc is true, the byte at the memory location specified by the contents of the Stack Pointer (SP) register pair is moved to the low order eight bits of the Program Counter (PC). The SP is 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 incremented again. The next Op Code following this instruction is fetched from the memory location specified by the PC. This instruction is normally used to return to the main line program 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 status that correspond to condition bits in the Flag Register (register F). These eight status are defined in the table below, which also specifies the corresponding cc bit fields in the assembled object code.

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

If cc is true:

M Cycles	T States	4 MHz E.T.
3	11 (5, 3, 3)	2.75

If cc is false:

M Cycles	T States	4 MHz E.T.
1	5	1.25

Condition Bits Affected: None

Example: If the S flag in the F register is set, the contents of the Program Counter are 3535H, the contents of the Stack Pointer are 2000H, the contents of memory location 2000H are B5H, and the contents of memory location 2001H are 18H. At execution of `RET M` the contents of the Stack Pointer is 2002H, and the contents of the Program Counter is 18B5H, pointing to the address of the next program Op Code to be fetched.

RETI

Operation: Return from Interrupt

Op Code: RETI

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

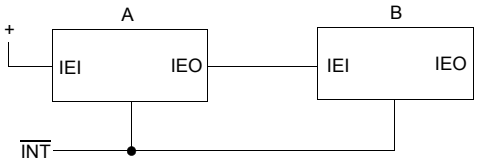
Description: This instruction is used at the end of a maskable interrupt service routine to:

- Restore the contents of the Program Counter (PC) (analogous to the RET instruction)
- Signal an I/O device that the interrupt routine is completed. The `RETI` instruction also facilitates the nesting of interrupts, allowing higher priority devices to temporarily suspend service of lower priority service routines. However, this instruction does not enable interrupts that 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 current service routine.

M Cycles	T States	4 MHz E.T.
4	14 (4, 4, 3, 3)	3.50

Condition Bits Affected: None

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



B generates an interrupt and is acknowledged. The interrupt enable out, IEO, of B goes Low, blocking any lower priority devices from interrupting while B is being serviced. Then A generates an interrupt, suspending service of B. The IEO of A goes Low, indicating that a higher priority device is being serviced. The A routine is completed and a `RETI` is issued resetting the IEO of A, allowing the B routine to continue. A second `RETI` is issued on completion of the B routine and the IE0 of B is reset (high) allowing lower priority devices interrupt access.

RETN

Operation: Return from non maskable interrupt

Op Code: RETN

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 interrupts service routine to restore the contents of the Program Counter (PC) (analogous to the RET instruction). The state of IFF2 is copied back to IFF1 so that maskable interrupts are enabled immediately following the RETN if they were enabled before the nonmaskable interrupt.

M Cycles	T States	4 MHz E.T.
4	14 (4, 4, 3, 3)	3.50

Condition Bits Affected: None

Example: If the contents of the Stack Pointer are `1000H`, and the contents of the Program Counter are `1A45H`, when a non maskable interrupt (NMI) signal is received, the CPU ignores the next instruction and instead restarts to memory address `0066H`. The current Program Counter contents of `1A45H` is pushed onto the external stack address of `0FFFH` and `0FFEH`, high order-byte first, and `0066H` is loaded onto the Program Counter. That address begins an interrupt service routine that 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 Op Code fetch to address `1A45H`, order-byte first, and 0066H is loaded onto the Program Counter. That address begins an interrupt service routine that ends with a `RETN` instruction. At 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

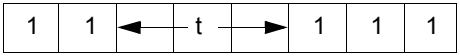
1000H. The program flow continues where it left off with an Op Code fetch to address 1A45H.

RST p

Operation: (SP-1) ← PCH, (SP-2) ← PCL, PCH ← 0, PCL ← P

Op Code: RST

Operands: p



Description: The current Program Counter (PC) contents are pushed onto the external memory stack, and the page zero memory location given by operand p is loaded to the PC. Program execution then begins with the Op Code 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 to the memory address now pointed to by SP, decrementing SP again, and loading the low order byte of PC to the address now pointed to by SP. The Restart instruction allows for a jump to one of eight addresses indicated in the table below. The operand p is assembled to the object code using the corresponding T state.

Because all addresses are in page zero of memory, the high order byte of PC is loaded with 00H. The number selected from the p column of the table is loaded to the low order byte of PC.

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

M Cycles	T States	4 MHz E.T.
3	11 (5, 3, 3)	2.75

Example: If the contents of the Program Counter are 15B3H, at execution of `RST 18H` (Object code `1101111`) the PC contains `0018H`, as the address of the next Op Code fetched.

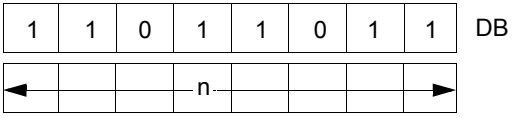
Input and Output Group

IN A, (n)

Operation: A ← (n)

Op Code: IN

Operands: A, (n)



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

M Cycles	T States	4 MHz LT.
3	11 (4, 3, 4)	2.75

Condition Bits Affected: None

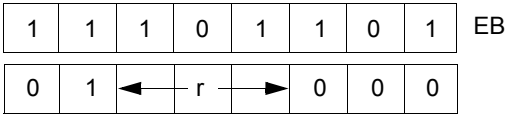
Example: If the contents of the Accumulator are `23H`, and byte `7BH` is available at the peripheral device mapped to I/O port address `01H`. At execution of `INA, (01H)` the Accumulator contains `7BH`.

IN r (C)

Operation: r ← (C)

Op Code: IN

Operands: r, (C)



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 to register r in the CPU. Register r identifies any of the CPU registers shown in the following table, which also indicates the corresponding 3-bit r field for each. The flags are affected, checking the input data.

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

M Cycles	T States	4 MHz E.T.
3	12 (4, 4, 4)	3.00

Condition Bits Affected:

S is set if input data is negative; reset otherwise
Z is set if input data is zero; reset otherwise
H is reset
P/V is set if parity is even; reset otherwise
N is reset
C is not affected

Example: If the contents of register C are `07H`, the contents of register B are `10H`, and byte `7BH` is available at the peripheral device mapped to I/O port address `07H`. After execution of `IN D, (C)` register D contains `7BH`.

INI

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

Op Code: INI

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

M Cycles	T States	4 MHz E.T.
4	16 (4, 5, 3, 4)	4.00

Condition Bits Affected:

S is unknown
Z is set if B–1 = 0, reset otherwise
H is unknown
P/V is unknown
N is set
C is 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 byte `7BH` is available at the peripheral device mapped to I /O port address `07H`. At execution of `INI` memory location `1000H` contains `7BH`, the HL register pair contains `1001H`, and register B contains `0FH`.

INIR

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

Op Code: INIR

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 from the selected port is placed on the data bus and written to the CPU. The contents of the HL register pair are placed on the address bus and the input byte is written to the corresponding location of memory. Then register pair HL is incremented, the byte counter is decremented. If decrementing causes B to go to zero, the instruction is terminated. If B is not zero, the PC is decremented by two and the instruction repeated. Interrupts are recognized and two refresh cycles execute after each data transfer.

Note: if B is set to zero prior to instruction execution, 256 bytes of data are input.

If B ≠ 0:

M Cycles	T States	4 MHz E.T.
5	21 (4, 5, 3, 4, 5)	5.25

If B = 0:

M Cycles	T States	4 MHz E.T.
4	16 (4, 5, 3, 4)	4.00

Condition Bits Affected:

S is unknown
Z is set
H is unknown
P/V is unknown

N is set
C is not affected

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

```
51H
A9H
03H
```

then at execution of `INIR` the HL register pair contains 1003H, register B contains zero, and memory locations contain the following:

`1000H`	contains `51H`
`1001H`	contains `A9H`
`1002H`	contains `03H`

IND

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

Op Code: IND

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

M Cycles	T States	4 MHz E.T.
4	16 (4, 5, 3, 4)	4.00

Condition Bits Affected:

S is unknown
Z is set if B–1 = 0; reset otherwise
H is unknown
P/V is unknown
N is set
C is 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 byte `7BH` is available at the peripheral device mapped to I/O port address `07H`. At execution of `IND` memory location `1000H` contains `7BH`, the HL register pair contains `0FFFH`, and register B contains `0FH`.

INDR

Operation: (HL) ← (C), B ← 131, HL ← HL1

Op Code: INDR

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 from the selected port is placed on the data bus and written to the CPU. The contents of the HL register pair are placed on the address bus and the input byte is written 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 PC is decremented by two and the instruction repeated. Interrupts are recognized and two refresh cycles are executed after each data transfer.

When B is set to zero prior to instruction execution, 256 bytes of data are input.

If B ≠ 0

M Cycles	T States	4 MHz E.T.
5	21 (4, 5, 3, 4, 5)	5.25

If B = 0:

M Cycles	T States	4 MHz E.T.
4	16 (4, 5, 3, 4)	4.00

Condition Bits Affected:

S is unknown
Z is set
H is unknown
P/V is unknown
N is set

C is not affected

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

```
51H
A9H
03H
```

then at execution of INDR the HL register pair contains 0FFDH, register B contains zero, and memory locations contain the following:

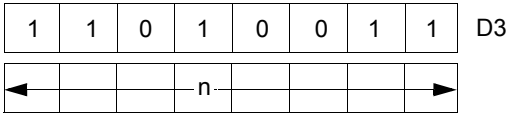
```
0FFEH    contains 03H
0FFFH    contains A9H
1000H    contains 51H
```

OUT (n), A

Operation: (n) ← A

Op Code: OUT

Operands: (n), A



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

M Cycles	T States	4 MHz E.T.
3	11 (4, 3, 4)	2.75

Condition Bits Affected: None

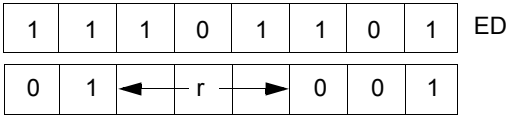
Example: If the contents of the Accumulator are `23H`, at execution of `OUT (01H)`, byte `23H` is written to the peripheral device mapped to I/O port address `01H`.

OUT (C), r

Operation: (C) ← r

Op Code: OUT

Operands: (C), r



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

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

M Cycles	T States	4 MHz E.T.
3	12 (4, 4, 4)	3.00

Condition Bits Affected: None

Example: If the contents of register C are `01H`, and the contents of register D are `5AH`, at execution of `OUT (C),D` byte `5AH` is written to the peripheral device mapped to I/O port address `01H`.

OUTI

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

Op Code: OUTI

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 (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. Register B may be used as a byte counter, and its decremented value is placed on the top half (A8 through A15) of the address bus. The byte to be output is placed on the data bus and written to a selected peripheral device. Finally, the register pair HL is incremented.

M Cycles	T States	4 MHz E.T.
4	16 (4, 5, 3, 4)	4.00

Condition Bits Affected:

S is unknown
Z is set if B–1 = 0; reset otherwise
H is unknown
P/V is unknown
N is set
C is 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 100014, and the contents of memory address `1000H` are 5914, then after thee execution of `OUTI` register B contains `0FH`, the HL register pair contains `1001H`, and byte `59H` is written to the peripheral device mapped to I/O port address `07H`.

OTIR

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

Op Code: OTIR

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 temporarily stored in the CPU. Then, after the byte counter (B) is decremented, the contents of register C are placed on the bottom half (A0 through A7) of the address bus to select the I/O device at one of 256 possible ports. Register B may be used as a byte counter, and its decremented value is placed on the top half (A8 through A15) of the address bus at this time. Next, the byte to be output is placed on the data bus and written to 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 are recognized and two refresh cycles are executed after each data transfer.

Note: When B is set to zero prior to instruction execution, the instruction outputs 256 bytes of data.

If B ≠ 0:

M Cycles	T States	4 MHz E.T.
5	21 (4, 5, 3, 4, 5)	5.25

If B = 0:

M Cycles	T States	4 MHz E.T.
4	16 (4, 5, 3, 4)	4.00

Condition Bits Affected:

S is unknown
Z is set
H is unknown
P/V is unknown
N is set
C is 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:

```
1000H    contains 51H
1001H    contains A9H
1002H    contains 03H
```

then at execution of OTIR the HL register pair contains 1003H, register B contains zero, and a group of bytes is written to the peripheral device mapped to I/O port address 07H in the following sequence:

```
51H
A9H
03H
```

OUTD

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

Op Code: OUTD

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

M Cycles	T States	4 MHz E.T.
4	16 (4, 5, 3. 4)	4.00

Condition Bits Affected:

S is unknown
Z is set if B–1 = 0; reset otherwise
H is unknown
P/V is unknown
N is set
C is 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`, at execution of `OUTD` register B contains `0FH`, the HL register pair contains `0FFFH`, and byte `59H` is written to the peripheral device mapped to I/O port address `07H`.

OTDR

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

Op Code: OTDR

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

Note: When B is set to zero prior to instruction execution, the instruction outputs 256 bytes of data.

If B ≠ 0:

M Cycles	T States	4 MHz E.T.
5	21 (4, 5, 3, 4, 5)	5.25

If B = 0:

M Cycles	T States	4 MHz E.T.
4	16 (4, 5, 3, 4)	4.00