

Zilog Z80 CPU Specifications

by Sean Young (syoun@cs.vu.nl)

Last update: September 21, 1997

Opcode Prefixes

There are four different opcode prefixes: CB, DD, ED and FD. If an opcode begins the DD prefix, the IX register is used in stead of the HL register and likewise the FD prefix results in use of the IY register in stead of the HL register. If HL refers to memory address, (e.g. LD A, (HL)), an offset d is added to the opcode (e.g. LD A, (IX + d)). If the instruction uses the H or L register, the high order byte or the low order byte of the IX or IY register is used. There are two exceptions to this rule, though: the EXX and EX DE, HL instructions. These instructions cannot be changed with a DD or FD prefix. If the instruction does not use the HL, H or L register, the instruction is executed as without the DD or FD prefix (except for the DDCB and FDCB opcode prefixes). Multiple DDs or FDs after each other operate like NOPs. If a instruction is preceded by an ED, a new set of instructions is used (see Opcode list). If the opcode is not listed, it will operate as two NOPs. Similarly the CB, FDCB and the DDCB select a new instruction set, though every possible instruction has a function and is listed.

The IM 0/1 and RETI/N instructions

The opcode list shows these two strange instructions. They are unofficial, and because both possibilities operate exactly the same way on a MSX, I can not determine which one they are. It is not important on a MSX though, but if anyone knows which one they are, please let me know.

Registers

The Z80 has the following (accessible) registers:

I, R, A, F, BC, DE, HL, IX, IY, SP, PC, AF', BC', DE' and HL'.

The R register is for memory refresh. It is increased by 1 after every instruction, whereby the CB, DD, DDCB, ED, FD and FDCB prefixes are viewed as separate instructions. Bit 7 is never modified, but it can be changed with LD A,R.

The F register is the flag register. It has the following structure:

- Bit 7: The S flag: if the result of the operation is negative, this flag is high. This is a copy of the MSB of the result of the operation.
 - Bit 6: The Z flag: if the result is zero, this flag is high.
 - Bit 5: This bit has no official name. Throughout this document it is referred to as 'b5'. It is a copy of bit 5 of the result of the operation.
 - Bit 4: The H flag. This is the carry from bit 3 to bit 4 of the operation. Used with DAA instruction.
 - Bit 3: This bit has no official name. Throughout this document it is referred to as 'b3'. It is a copy of bit 3 of the result of the operation.
 - Bit 2: The P/V flag. It contains the overflow (high if two's complements result does not fit in register) or the parity (parity of number of high bits in result of operation). See description of instructions for which one.
 - Bit 1: The N flag. It is high if the last operation was an subtraction, otherwise it was an addition. Used with DAA instruction.
 - Bit 0: The C flag. If the result of the operation does not fit in the register, this bit is high.
- Sometimes the flags have other meanings. See descriptions of instructions for details.

Interrupts

There are two types of interrupts: mask-able and non mask-able, and there are two flip-flops associated with interrupts: IFF₁ and IFF₂. DI resets both and EI sets both. If IFF₁ is set then mask-able interrupts are accepted. When a non mask-able interrupt occurs, the IFF₁ is reset, so disabling mask-able interrupts. When the CPU returns from a non mask-able interrupt (with RETN) the IFF₂ is copied into IFF₁ (and thus restored). Only IFF₂ can be read (with LD A, I and LD A, R IFF₂ is copied into the P/V flag).

MSX Specific: non mask-able interrupts never occur in a MSX, so the two interrupt flip-flops are always identical. Viewing the CPU as though it has one Interrupt flip-flop is not incorrect.

Interrupt Modes

Interrupt modes can be set with the IM x instructions. They only affect mask-able interrupts. When a mask-able interrupt occurs, the interrupting device must supply a value.

Interrupt Mode 0:

The value the interrupting device supplies is interpreted as an 8 bit opcode which is executed (usually RST p instructions).

Interrupt Mode 1:

A call is made to address 38h. The value the interrupting device supplies is ignored.

Interrupt Mode 2:

A call is made to an address read from address (register I × 256 + value from interrupting device).

MSX Specific: The MSX has one interrupting device (the VDP), which always provides value FFh (instruction RST 38h), so the MSX operates the same in Interrupt Mode 0 as in 1. In IM 2 a call is made to an address read from address (register I × 256 + FFh).

Instructions Descriptions

8 bit Load Group

Mnemonic	Symbolic Operation	S	Z	F5	H	F3	P/V	N	C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD r, r'	$r \leftarrow r'$	•	•	•	•	•	•	•	•	01 r r'		1	1	4	<u>r, r' Reg.</u>
LD p, p*	$p \leftarrow p'$	•	•	•	•	•	•	•	•	11 011 101 01 p p'	DD	2	2	8	000 B 001 C
LD q, q*	$q \leftarrow q'$	•	•	•	•	•	•	•	•	11 111 101 01 q q'	FD	2	2	8	010 D 011 E
LD r, n	$r \leftarrow n$	•	•	•	•	•	•	•	•	00 r 110 $\leftarrow n \rightarrow$		2	2	7	100 H 101 L
LD p, n*	$p \leftarrow n$	•	•	•	•	•	•	•	•	11 011 101 00 p 110 $\leftarrow n \rightarrow$	DD	3	3	11	111 A
LD q, n*	$q \leftarrow n$	•	•	•	•	•	•	•	•	11 111 101 00 q 110 $\leftarrow n \rightarrow$	FD	3	3	11	<u>p, p' Reg.</u> 000 B 001 C 010 D 011 E
LD r, (HL)	$r \leftarrow (HL)$	•	•	•	•	•	•	•	•	01 r 110		1	2	7	011 E
LD r, (IX + d)	$r \leftarrow (IX + d)$	•	•	•	•	•	•	•	•	11 011 101 01 r 110 $\leftarrow d \rightarrow$	DD	3	5	19	100 IX _H 101 IX _L 111 A
LD r, (IY + d)	$r \leftarrow (IY + d)$	•	•	•	•	•	•	•	•	11 111 101 01 r 110 $\leftarrow d \rightarrow$	FD	3	5	19	<u>q, q' Reg.</u> 000 B 001 C 010 D 011 E 100 IY _H 101 IY _L 111 A
LD (HL), r	$(HL) \leftarrow r$	•	•	•	•	•	•	•	•	01 110 r		1	2	7	001 C
LD (IX + d), r	$(IX + d) \leftarrow r$	•	•	•	•	•	•	•	•	11 011 101 01 110 r $\leftarrow d \rightarrow$	DD	3	5	19	010 D 011 E 100 IY _H 101 IY _L 111 A
LD (IY + d), r	$(IY + d) \leftarrow r$	•	•	•	•	•	•	•	•	11 111 101 01 110 r $\leftarrow d \rightarrow$	FD	3	5	19	
LD (HL), n	$(HL) \leftarrow n$	•	•	•	•	•	•	•	•	00 110 110 $\leftarrow n \rightarrow$	36	2	3	10	
LD (IX + d), n	$(IX + d) \leftarrow n$	•	•	•	•	•	•	•	•	11 011 101 00 110 110 $\leftarrow d \rightarrow$	DD 36	4	5	19	
LD (IY + d), n	$(IY + d) \leftarrow n$	•	•	•	•	•	•	•	•	11 111 101 00 110 110 $\leftarrow d \rightarrow$	FD 36	4	5	19	
LD A, (BC)	$A \leftarrow (BC)$	•	•	•	•	•	•	•	•	00 001 010 $\leftarrow n \rightarrow$	0A	1	2	7	
LD A, (DE)	$A \leftarrow (DE)$	•	•	•	•	•	•	•	•	00 011 010 $\leftarrow n \rightarrow$	1A	1	2	7	
LD A, (nn)	$A \leftarrow (nn)$	•	•	•	•	•	•	•	•	00 111 010 $\leftarrow n \rightarrow$	3A	3	4	13	
LD (BC), A	$(BC) \leftarrow A$	•	•	•	•	•	•	•	•	00 000 010 $\leftarrow n \rightarrow$	02	1	2	7	
LD (DE), A	$(DE) \leftarrow A$	•	•	•	•	•	•	•	•	00 010 010 $\leftarrow n \rightarrow$	12	1	2	7	
LD (nn), A	$(nn) \leftarrow A$	•	•	•	•	•	•	•	•	00 110 010 $\leftarrow n \rightarrow$	32	3	4	13	
LD A, I	$A \leftarrow I$	β	β	β	0	β	IFF ₂	0	•	11 101 101 01 010 111 $\leftarrow n \rightarrow$	ED 57	2	2	9	
LD A, R	$A \leftarrow R$	β	β	β	0	β	IFF ₂	0	•	11 101 101 01 011 111 $\leftarrow n \rightarrow$	ED 5F	2	2	9	R is read after it is increased.
LD I, A	$I \leftarrow A$	•	•	•	•	•	•	•	•	11 101 101 01 000 111 $\leftarrow n \rightarrow$	ED 47	2	2	9	
LD R, A	$R \leftarrow A$	•	•	•	•	•	•	•	•	11 101 101 01 001 111 $\leftarrow n \rightarrow$	ED 4F	2	2	9	R is written after it is increased.

Notes:

r, r' means any of the registers A, B, C, D, E, H, L.
p, p' means any of the registers A, B, C, D, E, IX_H, IX_L.
q, q' means any of the registers A, B, C, D, E, IY_H, IY_L.
dd_L, dd_H refer to high order and low order eight bits of the register respectively.
* means unofficial instruction.

Flag Notation:

• = flag is not affected, 0 = flag is reset, 1 = flag is set,
β = flag is set according to the result of the operation, IFF₂ = the interrupt flip-flop 2 is copied.

16 bit Load Group

Mnemonic	Symbolic Operation	Flags								Opcode			Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments	
		S	Z	F5	H	F3	P/V	N	C	76	543	210						
LD dd, nn	dd ← nn	•	•	•	•	•	•	•	•	00	dd0	001		3	3	10	dd Pair 00 BC 01 DE	
LD IX, nn	IX ← nn	•	•	•	•	•	•	•	•	← n →	11	011	101	DD	4	4	14	02 HL 03 SP
										00	110	001						
										21								
										← n →								
LD IY, nn	IY ← nn	•	•	•	•	•	•	•	•	← n →	11	111	101	FD	4	4	14	
										00	110	001						
										21								
										← n →								
LD HL, (nn)	L ← (nn) H ← (nn+1)	•	•	•	•	•	•	•	•	← n →	00	101	010	2A	3	5	16	
										← n →								
LD dd, (nn)	dd _L ← (nn) dd _H ← (nn+1)	•	•	•	•	•	•	•	•	← n →	11	101	101	ED	4	6	20	
										01	dd1	011						
										← n →								
LD IX, (nn)	IX _L ← (nn) IX _H ← (nn+1)	•	•	•	•	•	•	•	•	← n →	11	011	101	DD	4	6	20	
										00	101	010						
										2A								
LD IY, (nn)	IY _L ← (nn) IY _H ← (nn+1)	•	•	•	•	•	•	•	•	← n →	11	111	101	FD	4	6	20	
										00	101	010						
										2A								
LD (nn), HL	(nn) ← L (nn+1) ← H	•	•	•	•	•	•	•	•	← n →	00	100	010	22	3	5	16	
										← n →								
										← n →								
LD (nn), dd	(nn) ← dd _L (nn+1) ← dd _H	•	•	•	•	•	•	•	•	← n →	11	101	101	DD	4	6	20	
										01	dd0	011						
										← n →								
LD (nn), IX	(nn) ← IX _L (nn+1) ← IX _H	•	•	•	•	•	•	•	•	← n →	11	011	101	DD	4	6	20	
										00	100	010						
										22								
LD (nn), IY	(nn) ← IY _L (nn+1) ← IY _H	•	•	•	•	•	•	•	•	← n →	11	111	101	FD	4	6	20	
										00	100	010						
										22								
LD SP, HL	SP ← HL	•	•	•	•	•	•	•	•	← n →	11	111	001	F9	1	1	6	
										DD								
LD SP, IX	SP ← IX	•	•	•	•	•	•	•	•	← n →	11	011	101	DD	2	2	10	
										F9								
LD SP, IY	SP ← IY	•	•	•	•	•	•	•	•	← n →	11	111	101	FD	2	2	10	
										F9								
PUSH qq	SP ← SP - 1 (SP) ← qq _H SP ← SP - 1 (SP) ← qq _L	•	•	•	•	•	•	•	•	← n →	11	qq0	101		1	3	11	qq Pair 00 BC 01 DE 10 HL
										← n →								
										← n →								
										← n →								
PUSH IX	SP ← SP - 1 (SP) ← IX _H SP ← SP - 1 (SP) ← IX _L	•	•	•	•	•	•	•	•	← n →	11	011	101	DD	2	4	15	11 AF
										← n →								
										← n →								
										← n →								
PUSH IY	SP ← SP - 1 (SP) ← IY _H SP ← SP - 1 (SP) ← IY _L	•	•	•	•	•	•	•	•	← n →	11	111	101	FD	2	4	15	
										← n →								
										← n →								
										← n →								
POP qq	(SP) ← qq _L SP ← SP + 1 (SP) ← qq _H SP ← SP + 1	•	•	•	•	•	•	•	•	← n →	11	qq0	001		1	3	10	
										← n →								
										← n →								
										← n →								
POP IX	(SP) ← IX _L SP ← SP + 1 (SP) ← IX _H SP ← SP + 1	•	•	•	•	•	•	•	•	← n →	11	011	101	DD	2	4	14	
										← n →								
										← n →								
										← n →								
POP IY	(SP) ← IY _L SP ← SP + 1 (SP) ← IY _H SP ← SP + 1	•	•	•	•	•	•	•	•	← n →	11	111	101	FD	2	4	14	
										← n →								
										← n →								
										← n →								

Notes: dd is any of the register pair BC, DE, HL, SP.

qq is any of the register pair BC, DE, HL, AF.

Flag Notation: • = flag is not affected, 0 = flag is reset, 1 = flag is set, β = flag is set according to the result of the operation.

Exchange, Block Transfer and Search Groups

[illegible]

8 bit Arithmetic and Logical Group

Mnemonic	Symbolic Operation	Flags								Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
		S	Z	F5	H	F3	P/V	N	C						
ADD A, r	$A \leftarrow A + r$	β	β	β	β	β	V	0	β	10 000 r		1	1	4	r Reg. p Reg.
ADD A, p*	$A \leftarrow A + p$	β	β	β	β	β	V	0	β	11 011 101 10 000 p	DD	2	2	8	000 B 000 B 001 C 001 C
ADD A, q*	$A \leftarrow A + q$	β	β	β	β	β	V	0	β	11 111 101 10 000 q	FD	2	2	8	010 D 010 D 011 E 011 E
ADD A, n	$A \leftarrow A + n$	β	β	β	β	β	V	0	β	11 000 110 $\leftarrow n \rightarrow$		2	2	8	100 H 100 IX _H 101 L 101 IX _H
ADD A, (HL)	$A \leftarrow A + (HL)$	β	β	β	β	β	V	0	β	10 000 110		1	2	7	111 A 111 A
ADD A, (IX + d)	$A \leftarrow A + (IX + d)$	β	β	β	β	β	V	0	β	11 011 101 10 000 110	DD	3	5	19	
ADD A, (IY + d)	$A \leftarrow A + (IY + d)$	β	β	β	β	β	V	0	β	$\leftarrow d \rightarrow$ 11 111 101 10 000 110 $\leftarrow d \rightarrow$	FD	3	5	19	
ADC A, s	$A \leftarrow A + s + CY$	β	β	β	β	β	V	0	β	001					s is any of r, n, (HL), (IX+d), (IY+d), p, q as shown for the ADD instruction. The underlined bits replace the underlined bits in the ADD set.
SUB A, s	$A \leftarrow A - s$	β	β	β	β	β	V	1	β	010					
SBC A, s	$A \leftarrow A - s - CY$	β	β	β	β	β	V	1	β	011					
AND s	$A \leftarrow A \text{ AND } s$	β	β	β	1	β	P	0	0	100					
OR s	$A \leftarrow A \text{ OR } s$	β	β	β	0	β	P	0	0	110					
XOR s	$A \leftarrow A \text{ XOR } s$	β	β	β	0	β	P	0	0	101					q Reg. 000 B 001 C 010 D 011 E 100 IY _H 101 IY _L 111 A
CP s	$A - s$	β	β	β^1	β	β^1	V	1	β	111					
INC r	$r \leftarrow r + 1$	β	β	β	β	β	V	0	•	00 r 100		1	1	4	
INC p*	$p \leftarrow p + 1$	β	β	β	β	β	V	0	•	11 011 101 00 p 100	DD	2	2	8	
INC q*	$q \leftarrow q + 1$	β	β	β	β	β	V	0	•	11 111 101 00 q 100	FD	2	2	8	
INC (HL)	$(HL) \leftarrow (HL) + 1$	β	β	β	β	β	V	0	•	00 110 100		1	3	11	011 E
INC (IX + d)	$(IX + d) \leftarrow (IX + d) + 1$	β	β	β	β	β	V	0	•	11 011 101 00 110 100	DD	3	6	23	100 IY _H 101 IY _L 111 A
INC (IY + d)	$(IY + d) \leftarrow (IY + d) + 1$	β	β	β	β	β	V	0	•	$\leftarrow d \rightarrow$ 11 111 101 00 110 100 $\leftarrow d \rightarrow$	FD	3	6	23	
DEC m	$m \leftarrow m - 1$	β	β	β	β	β	V	1	•	101					m is any of r, p, q, (HL), (IX+d), (IY+d), as shown for the INC instruction. DEC same format and states as INC. Replace 100 with 101 in opcode.

Notes:

¹ F5 and F3 are copied from the operand (s), not from the result of (A - s).

The V symbol in the P/V flag column indicates that the P/V flags contains the overflow of the operation. Similarly the P symbol indicates parity.

r means any of the registers A, B, C, D, E, H, L.

p means any of the registers A, B, C, D, E, IX_H, IX_L.

q means any of the registers A, B, C, D, E, IY_H, IY_L.

dd_L, dd_H refer to high order and low order eight bits of the register respectively.

CY means the carry flip-flop.

* means unofficial instruction.

Flag Notation:

• = flag is not affected, 0 = flag is reset, 1 = flag is set, β = flag is set according to the result of the operation.

16 bit Arithmetic Group

Mnemonic	Symbolic Operation	Flags								Opcode				Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
		S	Z	F5	H	F3	P/V	N	C	76	54	3	210					
ADD HL, ss	HL ← HL + ss	•	•	β ²	β ²	β ²	•	0	β ¹	00	ss	1	001		1	3	11	ss Reg.
ADC HL, ss	HL ← HL + ss + CY	β ¹	β ¹	β ²	β ²	β ²	V ¹	0	β ¹	11	101	101		ED	2	4	15	00 BC 01 DE 10 HL 11 SP
SBC HL, ss	HL ← HL - ss - CY	β ¹	β ¹	β ²	β ²	β ²	V ¹	1	β ¹	01	ss	1	010	ED	2	4	15	10 HL 11 SP
ADD IX, pp	IX ← IX + pp	•	•	β ²	β ²	β ²	•	0	β ¹	11	011	101		DD	2	4	15	pp Reg.
ADD IY, rr	IY ← IY + rr	•	•	β ²	β ²	β ²	•	0	β ¹	11	111	101		FD	2	4	15	00 BC 01 DE 10 IX 11 SP
INC ss	ss ← ss + 1	•	•	•	•	•	•	•	•	00	ss	0	111		1	1	6	10 IX
INC IX	IX ← IX + 1	•	•	•	•	•	•	•	•	11	011	101		DD	2	2	10	11 SP
INC IY	IY ← IY + 1	•	•	•	•	•	•	•	•	00	100	011						
DEC ss	ss ← ss - 1	•	•	•	•	•	•	•	•	11	111	101		FD	2	2	10	rr Reg.
DEC IX	IX ← IX - 1	•	•	•	•	•	•	•	•	00	100	011						00 BC
DEC IY	IY ← IY - 1	•	•	•	•	•	•	•	•	11	111	101		FD	2	2	10	01 DE
		•	•	•	•	•	•	•	•	00	101	011		2B				10 IY
		•	•	•	•	•	•	•	•	11	111	101		FD	2	2	10	11 SP
		•	•	•	•	•	•	•	•	00	101	011		2B				

Notes: The V symbol in the P/V flag column indicates that the P/V flags contains the overflow of the operation.
ss means any of the registers BC, DE, HL, SP.
pp means any of the registers BC, DE, IX, SP.
rr means any of the registers BC, DE, IY, SP.
16 bit additions are performed by first adding the two low order eight bits, and then the two high order eight bits.
¹ Indicates the flag is affected by the 16 bit result of the operation.
² Indicates the flag is affected by the 8 bit addition of the high order eight bits.
CY means the carry flip-flop.

Flag Notation: • = flag is not affected, 0 = flag is reset, 1 = flag is set, β = flag is set according to the result of the operation.

General Purpose Arithmetic and CPU Control Groups

Mnemonic	Symbolic Operation	Flags								Opcode				Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
		S	Z	F5	H	F3	P/V	N	C	76	54	3	210					
DAA	Converts A into packed BCD following add or subtract with BCD operands.	β	β	β	β	β	P	•	β	00	100	111		27	1	1	4	
CPL	A ← $\overline{A}$	•	•	β ¹	1	β ¹	•	1	•	00	101	111		2F	1	1	4	One's complement.
NEG ⁴	A ← 0 - A	β	β	β	β	β	V	1	β	11	101	101		ED	2	2	8	Two's complement.
CCF	CY ← $\overline{CY}$	•	•	β ¹	β ²	β ¹	•	0	β	00	111	111		3F	1	1	4	Complement carry flag.
SCF	CY ← 1	•	•	β ¹	0	β ¹	•	0	1	00	110	111		37	1	1	4	
NOP	No operations	•	•	•	•	•	•	•	•	00	000	000		00	1	1	4	
HALT	CPU halted	•	•	•	•	•	•	•	•	01	110	110		76	1	1	4	
DI ³	IFF ₁ ← 0	•	•	•	•	•	•	•	•	11	110	011		F3	1	1	4	
EI ³	IFF ₁ ← 1	•	•	•	•	•	•	•	•	11	111	011		FB	1	1	4	
IM 0 ⁴	Set interrupt mode 0	•	•	•	•	•	•	•	•	11	101	101		ED	2	2	8	
IM 1 ⁴	Set interrupt mode 1	•	•	•	•	•	•	•	•	01	000	110		46				
IM 2 ⁴	Set interrupt mode 2	•	•	•	•	•	•	•	•	11	101	101		ED	2	2	8	
		•	•	•	•	•	•	•	•	01	010	110		56				
		•	•	•	•	•	•	•	•	11	011	110		5E				

Notes: The V symbol in the P/V flag column indicates that the P/V flags contains the overflow of the operation. Similarly the P symbol indicates parity.

¹ F5 and F3 are a copy of bit 5 and 3 of register A

² H contains the previous carry state (after instruction H ↔ C)

³ No interrupts are issued directly after a DI or EI.

⁴ This instruction has other unofficial opcodes, see Opcodes list.

CY means the carry flip-flop.

Flag Notation: • = flag is not affected, 0 = flag is reset, 1 = flag is set, β = flag is set according to the result of the operation.

Rotate and Shift Group

Mnemonic	Symbolic Operation	Flags								Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
		S	Z	F5	H	F3	P/V	N	C						
RLCA		•	•	β	0	β	•	0	β	00 000 111	07	1	1	4	
RLA		•	•	β	0	β	•	0	β	00 010 111	17	1	1	4	
RRCA		•	•	β	0	β	•	0	β	00 001 111	0F	1	1	4	
RRA		•	•	β	0	β	•	0	β	00 011 111	1F	1	1	4	
RLC r		β	β	β	0	β	P	0	β	11 001 011 00 000 r	CB	2	2	8	r Reg. 000 B
RLC (HL)		β	β	β	0	β	P	0	β	11 001 011 00 000	CB	2	4	15	001 C 010 D
RLC (IX + d)		β	β	β	0	β	P	0	β	11 011 101 11 001 011	DD CB	4	6	23	011 E 100 H 101 L 111 A
										← d → 00 000 110					
RLC (IY + d)		β	β	β	0	β	P	0	β	11 111 101 11 001 011	FD CB	4	6	23	
										← d → 00 000 110					
LD r,RLC (IX + d)*	r ← (IX + d) RLC r (IX + d) ← r	β	β	β	0	β	P	0	β	11 011 101 11 001 011	DD CB	4	6	23	
										← d → 00 000 r 110					
LD r,RLC (IY + d)*	r ← (IY + d) RLC r (IY + d) ← r	β	β	β	0	β	P	0	β	11 111 101 11 001 011	FD CB	4	6	23	
										← d → 00 000 r 110					
RL m		β	β	β	0	β	P	0	β	010					Instruction format and states are the same as RLC. Replace 000 with new number.
RRC m		β	β	β	0	β	P	0	β	001					
RR m		β	β	β	0	β	P	0	β	011					
SLA m		β	β	β	0	β	P	0	β	100					
SLL m*		β	β	β	0	β	P	0	β	110					
SRA m		β	β	β	0	β	P	0	β	101					
SRL m		β	β	β	0	β	P	0	β	111					
RLD		β	β	β	0	β	P	0	•	11 101 101 01 101 111	ED 6F	2	5	18	
RRD		β	β	β	0	β	P	0	•	11 101 101 01 100 111	ED 67	2	5	18	

Notes: The P symbol in the P/V flag column indicates that the P/V flags contains the parity of the result.

r means any of the registers A, B, C, D, E, H, L.

* means unofficial instruction.

CY means the carry flip-flop.

Flag Notation: • = flag is not affected, 0 = flag is reset, 1 = flag is set, β = flag is set according to the result of the operation.

Bit Manipulation Group

[illegible]

Input and Output Groups

[illegible]

Jump Group

Mnemonic	Symbolic Operation	Flags								Opcode			Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
		S	Z	F5	H	F3	P/V	N	C	76	543	210					
JP nn	PC ← nn	•	•	•	•	•	•	•	•	11	000	011	C3	3	3	10	
										←	n	→					
										←	n	→					
JP cc, nn	if cc is true, PC ← nn	•	•	•	•	•	•	•	•	11	ccc	010		3	3	10	<u>ccc Condition</u>
										←	n	→					000 NZ
										←	n	→					001 Z
																	010 NC
																	011 C
																	100 PO
																	101 PE
																	110 P
																	111 M
JR e	PC ← PC + e	•	•	•	•	•	•	•	•	00	011	000	18	2	3	12	
										←	e - 2	→					
JR ss, e	if ss is true	•	•	•	•	•	•	•	•	00	ss	000		2	3	12	if ss is true
	PC ← PC + e									←	e - 2	→		2	2	7	if ss is false
JP HL	PC ← HL	•	•	•	•	•	•	•	•	11	101	001	E9	1	1	4	
JP IX	PC ← IX	•	•	•	•	•	•	•	•	11	011	101	DD	2	2	8	<u>ss Condition</u>
										11	101	001	E9				111 C
																	110 NC
JP IY	PC ← IY	•	•	•	•	•	•	•	•	11	111	101	FD	2	2	8	101 Z
										11	101	001	E9				100 NZ
DJNZ e	B ← B - 1	•	•	•	•	•	•	•	•	00	010	000	10	2	2	8	if B = 0
	if B ≠ 0									←	e - 2	→		2	3	13	if B ≠ 0
	PC ← PC + e																
Notes:	e is a signed two-complement number in the range <-126, 129>																
	e - 2 in the opcode provides an effective number of PC + e as PC incremented by 2 prior to the addition of e.																
Flag Notation:	• = flag is not affected, 0 = flag is reset, 1 = flag is set, β = flag is set according to the result of the operation.																

Call and Return Group

Mnemonic	Symbolic Operation	Flags								Opcode				No.of Bytes	No.of M Cycles	No.of T States	Comments
		S	Z	F5	H	F3	P/V	N	C	76	543	210	Hex				
CALL nn	SP ← SP - 1 (SP) ← PC _H SP ← SP - 1 (SP) ← PC _L PC ← nn	•	•	•	•	•	•	•	•	11	001	101	CD	3	5	17	
CALL cc, nn	if cc is true, SP ← SP - 1 (SP) ← PC _H SP ← SP - 1 (SP) ← PC _L PC ← nn	•	•	•	•	•	•	•	•	11	ccc	100		3	3	10	if cc is false
										←	n	→		3	5	17	if cc is true
										←	n	→					
RET	PC _L ← (SP) SP ← SP + 1 PC _H ← (SP) SP ← SP + 1	•	•	•	•	•	•	•	•	11	001	001	C9	1	3	10	
RET cc	if cc is true, PC _L ← (SP) SP ← SP + 1 PC _H ← (SP) SP ← SP + 1	•	•	•	•	•	•	•	•	11	ccc	000		1	1	5	if cc is false
														1	3	11	if cc is true
RETI ²	PC _L ← (SP) SP ← SP + 1 PC _H ← (SP) SP ← SP + 1	•	•	•	•	•	•	•	•	11	101	101	ED	2	4	14	<u>cc Condition</u>
										01	001	101	4D				000 NZ
																	001 Z
																	010 NC
																	011 C
RETN ^{1,2}	PC _L ← (SP) SP ← SP + 1 PC _H ← (SP) SP ← SP + 1 IFF ₁ ← IFF ₂	•	•	•	•	•	•	•	•	11	101	101	ED	2	4	14	
										01	000	101	45				100 PO
																	101 PE
																	110 P
																	111 M
RST p	SP ← SP - 1 (SP) ← PC _H SP ← SP - 1 (SP) ← PC _L PC ← p	•	•	•	•	•	•	•	•	11	t	111		1	3	11	<u>t</u> p
																	000 0h
																	001 8h
																	010 10h
																	011 18h
																	100 20h
																	101 28h
																	110 30h
																	111 38h

Notes: ¹ This instruction has other unofficial opcodes, see Opcode list.

² Instruction also IFF₁ ← IFF₂

Flag Notation: • = flag is not affected, 0 = flag is reset, 1 = flag is set, β = flag is set according to the result of the operation.

Instructions sorted by opcode

If an EDxx instruction is not listed, it should operate as two NOPs.

If a DDxx or FDxx instruction is not listed, it should operate as without the DD or FD prefix.

An asterisk (*) after a instruction means it is unofficial.

00	NOP	46	LD B, (HL)	8C	ADC A, H
01 n n	LD BC, nn	47	LD B, A	8D	ADC A, L
02	LD (BC), A	48	LD C, B	8E	ADC A, (HL)
03	INC BC	49	LD C, C	8F	ADC A, A
04	INC B	4A	LD C, D	90	SUB B
05	DEC B	4B	LD C, E	91	SUB C
06 n	LD B, n	4C	LD C, H	92	SUB D
07	RLCA	4D	LD C, L	93	SUB E
08	EX AF, AF'	4E	LD C, (HL)	94	SUB H
09	ADD HL, BC	4F	LD C, A	95	SUB L
0A	LD A, (BC)	50	LD D, B	96	SUB (HL)
0B	DEC BC	51	LD D, C	97	SUB A
0C	INC C	52	LD D, D	98	SBC A, B
0D	DEC C	53	LD D, E	99	SBC A, C
0E n	LD C, n	54	LD D, H	9A	SBC A, D
0F	RRC A	55	LD D, L	9B	SBC A, E
10 n	DJNZ PC + n	56	LD D, (HL)	9C	SBC A, H
11 n n	LD DE, nn	57	LD D, A	9D	SBC A, L
12	LD (DE), A	58	LD E, B	9E	SBC A, (HL)
13	INC DE	59	LD E, C	9F	SBC A, A
14	INC D	5A	LD E, D	A0	AND B
15	DEC D	5B	LD E, E	A1	AND C
16 n	LD D, n	5C	LD E, H	A2	AND D
17	RLA	5D	LD E, L	A3	AND E
18 n	JR PC + n	5E	LD E, (HL)	A4	AND H
19	ADD HL, DE	5F	LD E, A	A5	AND L
1A	LD A, (DE)	60	LD H, B	A6	AND (HL)
1B	DEC DE	61	LD H, C	A7	AND A
1C	INC E	62	LD H, D	A8	XOR B
1D	DEC E	63	LD H, E	A9	XOR C
1E n	LD E, n	64	LD H, H	AA	XOR D
1F	RRA	65	LD H, L	AB	XOR E
20 n	JR NZ, PC + n	66	LD H, (HL)	AC	XOR H
21 n n	LD HL, nn	67	LD H, A	AD	XOR L
22 n n	LD (nn), HL	68	LD L, B	AE	XOR (HL)
23	INC HL	69	LD L, C	AF	XOR A
24	INC H	6A	LD L, D	B0	OR B
25	DEC H	6B	LD L, E	B1	OR C
26 n	LD H, n	6C	LD L, H	B2	OR D
27	DAA	6D	LD L, L	B3	OR E
28 n	JR Z, PC + n	6E	LD L, (HL)	B4	OR H
29	ADD HL, HL	6F	LD L, A	B5	OR L
2A n n	LD HL, (nn)	70	LD (HL), B	B6	OR (HL)
2B	DEC HL	71	LD (HL), C	B7	ORA
2C	INC L	72	LD (HL), D	B8	CP B
2D	DEC L	73	LD (HL), E	B9	CP C
2E n	LD L, n	74	LD (HL), H	BA	CP D
2F	CPL	75	LD (HL), L	BB	CP E
30 n	JR NC, PC + n	76	HALT	BC	CP H
31 n n	LD SP, nn	77	LD (HL), A	BD	CP L
32 n n	LD (nn), A	78	LD A, B	BE	CP (HL)
33	INC SP	79	LD A, C	BF	CP A
34	INC (HL)	7A	LD A, D	C0	RET NZ
35	DEC (HL)	7B	LD A, E	C1	POP BC
36 n	LD (HL), n	7C	LD A, H	C2 n n	JP NZ, nn
37	SCF	7D	LD A, L	C3 n n	JP nn
38 n	JR C, PC + n	7E	LD A, (HL)	C4 n n	CALL NZ, nn
39	ADD HL, SP	7F	LD A, A	C5	PUSH BC
3A n n	LD A, (nn)	80	ADD A, B	C6 n	ADD A, n
3B	DEC SP	81	ADD A, C	C7	RST 0h
3C	INC A	82	ADD A, D	C8	RET Z
3D	DEC A	83	ADD A, E	C9	RET
3E n	LD A, n	84	ADD A, H	CA n n	JP Z, nn
3F	CCF	85	ADD A, L	CB00	RLC B
40	LD B, B	86	ADD A, (HL)	CB01	RLC C
41	LD B, C	87	ADD A, A	CB02	RLC D
42	LD B, D	88	ADC A, B	CB03	RLC E
43	LD B, E	89	ADC A, C	CB04	RLC H
44	LD B, H	8A	ADC A, D	CB05	RLC L
45	LD B, L	8B	ADC A, E	CB06	RLC (HL)

CB07	RLC A	CB5A	BIT 3, D	CBAD	RES 5, L
CB08	RRC B	CB5B	BIT 3, E	CBAE	RES 5, (HL)
CB09	RRC C	CB5C	BIT 3, H	CBAF	RES 5, A
CB0A	RRC D	CB5D	BIT 3, L	CBB0	RES 6, B
CB0B	RRC E	CB5E	BIT 3, (HL)	CBB1	RES 6, C
CB0C	RRC H	CB5F	BIT 3, A	CBB2	RES 6, D
CB0D	RRC L	CB60	BIT 4, B	CBB3	RES 6, E
CB0E	RRC (HL)	CB61	BIT 4, C	CBB4	RES 6, H
CB0F	RRC A	CB62	BIT 4, D	CBB5	RES 6, L
CB10	RL B	CB63	BIT 4, E	CBB6	RES 6, (HL)
CB11	RL C	CB64	BIT 4, H	CBB7	RES 6, A
CB12	RL D	CB65	BIT 4, L	CBB8	RES 7, B
CB13	RL E	CB66	BIT 4, (HL)	CBB9	RES 7, C
CB14	RL H	CB67	BIT 4, A	CBBA	RES 7, D
CB15	RL L	CB68	BIT 5, B	CBBB	RES 7, E
CB16	RL (HL)	CB69	BIT 5, C	CBBC	RES 7, H
CB17	RL A	CB6A	BIT 5, D	CBBD	RES 7, L
CB18	RR B	CB6B	BIT 5, E	CBBE	RES 7, (HL)
CB19	RR C	CB6C	BIT 5, H	CBBF	RES 7, A
CB1A	RR D	CB6D	BIT 5, L	CBC0	SET 0, B
CB1B	RR E	CB6E	BIT 5, (HL)	CBC1	SET 0, C
CB1C	RR H	CB6F	BIT 5, A	CBC2	SET 0, D
CB1D	RR L	CB70	BIT 6, B	CBC3	SET 0, E
CB1E	RR (HL)	CB71	BIT 6, C	CBC4	SET 0, H
CB1F	RR A	CB72	BIT 6, D	CBC5	SET 0, L
CB20	SLA B	CB73	BIT 6, E	CBC6	SET 0, (HL)
CB21	SLA C	CB74	BIT 6, H	CBC7	SET 0, A
CB22	SLA D	CB75	BIT 6, L	CBC8	SET 1, B
CB23	SLA E	CB76	BIT 6, (HL)	CBC9	SET 1, C
CB24	SLA H	CB77	BIT 6, A	CBCA	SET 1, D
CB25	SLA L	CB78	BIT 7, B	CBCB	SET 1, E
CB26	SLA (HL)	CB79	BIT 7, C	CBCC	SET 1, H
CB27	SLA A	CB7A	BIT 7, D	CBCD	SET 1, L
CB28	SRA B	CB7B	BIT 7, E	CBCE	SET 1, (HL)
CB29	SRA C	CB7C	BIT 7, H	CBCF	SET 1, A
CB2A	SRA D	CB7D	BIT 7, L	CBD0	SET 2, B
CB2B	SRA E	CB7E	BIT 7, (HL)	CBD1	SET 2, C
CB2C	SRA H	CB7F	BIT 7, A	CBD2	SET 2, D
CB2D	SRA L	CB80	RES 0, B	CBD3	SET 2, E
CB2E	SRA (HL)	CB81	RES 0, C	CBD4	SET 2, H
CB2F	SRA A	CB82	RES 0, D	CBD5	SET 2, L
CB30	SLL B*	CB83	RES 0, E	CBD6	SET 2, (HL)
CB31	SLL C*	CB84	RES 0, H	CBD7	SET 2, A
CB32	SLL D*	CB85	RES 0, L	CBD8	SET 3, B
CB33	SLL E*	CB86	RES 0, (HL)	CBD9	SET 3, C
CB34	SLL H*	CB87	RES 0, A	CBDA	SET 3, D
CB35	SLL L*	CB88	RES 1, B	CBDB	SET 3, E
CB36	SLL (HL)*	CB89	RES 1, C	CBDC	SET 3, H
CB37	SLL A*	CB8A	RES 1, D	CBDD	SET 3, L
CB38	SRL B	CB8B	RES 1, E	CBDE	SET 3, (HL)
CB39	SRL C	CB8C	RES 1, H	CBDF	SET 3, A
CB3A	SRL D	CB8D	RES 1, L	CBE0	SET 4, B
CB3B	SRL E	CB8E	RES 1, (HL)	CBE1	SET 4, C
CB3C	SRL H	CB8F	RES 1, A	CBE2	SET 4, D
CB3D	SRL L	CB90	RES 2, B	CBE3	SET 4, E
CB3E	SRL (HL)	CB91	RES 2, C	CBE4	SET 4, H
CB3F	SRL A	CB92	RES 2, D	CBE5	SET 4, L
CB40	BIT 0, B	CB93	RES 2, E	CBE6	SET 4, (HL)
CB41	BIT 0, C	CB94	RES 2, H	CBE7	SET 4, A
CB42	BIT 0, D	CB95	RES 2, L	CBE8	SET 5, B
CB43	BIT 0, E	CB96	RES 2, (HL)	CBE9	SET 5, C
CB44	BIT 0, H	CB97	RES 2, A	CBEA	SET 5, D
CB45	BIT 0, L	CB98	RES 3, B	CBEB	SET 5, E
CB46	BIT 0, (HL)	CB99	RES 3, C	CBEC	SET 5, H
CB47	BIT 0, A	CB9A	RES 3, D	CBED	SET 5, L
CB48	BIT 1, B	CB9B	RES 3, E	CBEE	SET 5, (HL)
CB49	BIT 1, C	CB9C	RES 3, H	CBEF	SET 5, A
CB4A	BIT 1, D	CB9D	RES 3, L	CBF0	SET 6, B
CB4B	BIT 1, E	CB9E	RES 3, (HL)	CBF1	SET 6, C
CB4C	BIT 1, H	CB9F	RES 3, A	CBF2	SET 6, D
CB4D	BIT 1, L	CBA0	RES 4, B	CBF3	SET 6, E
CB4E	BIT 1, (HL)	CBA1	RES 4, C	CBF4	SET 6, H
CB4F	BIT 1, A	CBA2	RES 4, D	CBF5	SET 6, L
CB50	BIT 2, B	CBA3	RES 4, E	CBF6	SET 6, (HL)
CB51	BIT 2, C	CBA4	RES 4, H	CBF7	SET 6, A
CB52	BIT 2, D	CBA5	RES 4, L	CBF8	SET 7, B
CB53	BIT 2, E	CBA6	RES 4, (HL)	CBF9	SET 7, C
CB54	BIT 2, H	CBA7	RES 4, A	CBFA	SET 7, D
CB55	BIT 2, L	CBA8	RES 5, B	CBFB	SET 7, E
CB56	BIT 2, (HL)	CBA9	RES 5, C	CBFC	SET 7, H
CB57	BIT 2, A	CBAA	RES 5, D	CBFD	SET 7, L
CB58	BIT 3, B	CBAB	RES 5, E	CBFE	SET 7, (HL)
CB59	BIT 3, C	CBAC	RES 5, H	CBFF	SET 7, A

CC n n	CALL Z, nn	DD9D	SBC A, IX _L *	DDCB d 45	BIT 0, (IX + d)*
CD n n	CALL nn	DD9E d	SBC A, (IX + d)	DDCB d 46	BIT 0, (IX + d)
CE n	ADC A, n	DDA4	AND IX _H *	DDCB d 47	BIT 0, (IX + d)*
CF	RST 8h	DDA5	AND IX _L *	DDCB d 48	BIT 1, (IX + d)*
D0	RET NC	DDA6 d	AND (IX + d)	DDCB d 49	BIT 1, (IX + d)*
D1	POP DE	DDAC	XOR IX _H *	DDCB d 4A	BIT 1, (IX + d)*
D2 n n	JP NC, nn	DDAD	XOR IX _L *	DDCB d 4B	BIT 1, (IX + d)*
D3 n	OUT (n), A	DDAE d	XOR (IX + d)	DDCB d 4C	BIT 1, (IX + d)*
D4 n n	CALL NC, nn	DDB4	OR IX _H *	DDCB d 4D	BIT 1, (IX + d)*
D5	PUSH DE	DDB5	OR IX _L *	DDCB d 4E	BIT 1, (IX + d)
D6 n	SUB n	DDB6 d	OR (IX + d)	DDCB d 4F	BIT 1, (IX + d)*
D7	RST 10h	DDBC	CP IX _H *	DDCB d 50	BIT 2, (IX + d)*
D8	RETC	DDBD	CP IX _L *	DDCB d 51	BIT 2, (IX + d)*
D9	EXX	DDBE d	CP (IX + d)	DDCB d 52	BIT 2, (IX + d)*
DA n n	JP C, nn	DDCB d 00	LD B, RLC (IX + d)*	DDCB d 53	BIT 2, (IX + d)*
DB n	IN A, (n)	DDCB d 01	LD C, RLC (IX + d)*	DDCB d 54	BIT 2, (IX + d)*
DC n n	CALL C, nn	DDCB d 02	LD D, RLC (IX + d)*	DDCB d 55	BIT 2, (IX + d)*
DD09	ADD IX, BC	DDCB d 03	LD E, RLC (IX + d)*	DDCB d 56	BIT 2, (IX + d)
DD19	ADD IX, DE	DDCB d 04	LD H, RLC (IX + d)*	DDCB d 57	BIT 2, (IX + d)*
DD21 n n	LD IX, nn	DDCB d 05	LD L, RLC (IX + d)*	DDCB d 58	BIT 3, (IX + d)*
DD22 n n	LD (nn), IX	DDCB d 06	RLC (IX + d)	DDCB d 59	BIT 3, (IX + d)*
DD23	INC IX	DDCB d 07	LD A, RLC (IX + d)*	DDCB d 5A	BIT 3, (IX + d)*
DD24	INC IX _H *	DDCB d 08	LD B, RRC (IX + d)*	DDCB d 5B	BIT 3, (IX + d)*
DD25	DEC IX _H *	DDCB d 09	LD C, RRC (IX + d)*	DDCB d 5C	BIT 3, (IX + d)*
DD26 n	LD IX _H , n*	DDCB d 0A	LD D, RRC (IX + d)*	DDCB d 5D	BIT 3, (IX + d)*
DD29	ADD IX, IX	DDCB d 0B	LD E, RRC (IX + d)*	DDCB d 5E	BIT 3, (IX + d)
DD2A n n	LD IX, (nn)	DDCB d 0C	LD H, RRC (IX + d)*	DDCB d 5F	BIT 3, (IX + d)*
DD2B	DEC IX	DDCB d 0D	LD L, RRC (IX + d)*	DDCB d 60	BIT 4, (IX + d)*
DD2C	INC IX _L *	DDCB d 0E	RRC (IX + d)	DDCB d 61	BIT 4, (IX + d)*
DD2D	DEC IX _L *	DDCB d 0F	LD A, RRC (IX + d)*	DDCB d 62	BIT 4, (IX + d)*
DD2E n	LD IX _L , n*	DDCB d 10	LD B, RL (IX + d)*	DDCB d 63	BIT 4, (IX + d)*
DD34 d	INC (IX + d)	DDCB d 11	LD C, RL (IX + d)*	DDCB d 64	BIT 4, (IX + d)*
DD35 d	DEC (IX + d)	DDCB d 12	LD D, RL (IX + d)*	DDCB d 65	BIT 4, (IX + d)*
DD36 d n	LD (IX + d), n	DDCB d 13	LD E, RL (IX + d)*	DDCB d 66	BIT 4, (IX + d)
DD39	ADD IX, SP	DDCB d 14	LD H, RL (IX + d)*	DDCB d 67	BIT 4, (IX + d)*
DD44	LD B, IX _H *	DDCB d 15	LD L, RL (IX + d)*	DDCB d 68	BIT 5, (IX + d)*
DD45	LD B, IX _L *	DDCB d 16	RL (IX + d)	DDCB d 69	BIT 5, (IX + d)*
DD46 d	LD B, (IX + d)	DDCB d 17	LD A, RL (IX + d)*	DDCB d 6A	BIT 5, (IX + d)*
DD4C	LD C, IX _H *	DDCB d 18	LD B, RR (IX + d)*	DDCB d 6B	BIT 5, (IX + d)*
DD4D	LD C, IX _L *	DDCB d 19	LD C, RR (IX + d)*	DDCB d 6C	BIT 5, (IX + d)*
DD4E d	LD C, (IX + d)	DDCB d 1A	LD D, RR (IX + d)*	DDCB d 6D	BIT 5, (IX + d)*
DD54	LD D, IX _H *	DDCB d 1B	LD E, RR (IX + d)*	DDCB d 6E	BIT 5, (IX + d)
DD55	LD D, IX _L *	DDCB d 1C	LD H, RR (IX + d)*	DDCB d 6F	BIT 5, (IX + d)*
DD56 d	LD D, (IX + d)	DDCB d 1D	LD L, RR (IX + d)*	DDCB d 70	BIT 6, (IX + d)*
DD5C	LD E, IX _H *	DDCB d 1E	RR (IX + d)	DDCB d 71	BIT 6, (IX + d)*
DD5D	LD E, IX _L *	DDCB d 1F	LD A, RR (IX + d)*	DDCB d 72	BIT 6, (IX + d)*
DD5E d	LD E, (IX + d)	DDCB d 20	LD B, SLA (IX + d)*	DDCB d 73	BIT 6, (IX + d)*
DD60	LD IX _H , B*	DDCB d 21	LD C, SLA (IX + d)*	DDCB d 74	BIT 6, (IX + d)*
DD61	LD IX _H , C*	DDCB d 22	LD D, SLA (IX + d)*	DDCB d 75	BIT 6, (IX + d)*
DD62	LD IX _H , D*	DDCB d 23	LD E, SLA (IX + d)*	DDCB d 76	BIT 6, (IX + d)
DD63	LD IX _H , E*	DDCB d 24	LD H, SLA (IX + d)*	DDCB d 77	BIT 6, (IX + d)*
DD64	LD IX _H , IX _H *	DDCB d 25	LD L, SLA (IX + d)*	DDCB d 78	BIT 7, (IX + d)*
DD65	LD IX _H , IX _L *	DDCB d 26	SLA (IX + d)	DDCB d 79	BIT 7, (IX + d)*
DD66 d	LD H, (IX + d)	DDCB d 27	LD A, SLA (IX + d)*	DDCB d 7A	BIT 7, (IX + d)*
DD67	LD IX _H , A*	DDCB d 28	LD B, SRA (IX + d)*	DDCB d 7B	BIT 7, (IX + d)*
DD68	LD IX _L , B*	DDCB d 29	LD C, SRA (IX + d)*	DDCB d 7C	BIT 7, (IX + d)*
DD69	LD IX _L , C*	DDCB d 2A	LD D, SRA (IX + d)*	DDCB d 7D	BIT 7, (IX + d)*
DD6A	LD IX _L , D*	DDCB d 2B	LD E, SRA (IX + d)*	DDCB d 7E	BIT 7, (IX + d)
DD6B	LD IX _L , E*	DDCB d 2C	LD H, SRA (IX + d)*	DDCB d 7F	BIT 7, (IX + d)*
DD6C	LD IX _L , IX _H *	DDCB d 2D	LD L, SRA (IX + d)*	DDCB d 80	LD B, RES 0, (IX + d)*
DD6D	LD IX _L , IX _L *	DDCB d 2E	SRA (IX + d)	DDCB d 81	LD C, RES 0, (IX + d)*
DD6E d	LD L, (IX + d)	DDCB d 2F	LD A, SRA (IX + d)*	DDCB d 82	LD D, RES 0, (IX + d)*
DD6F	LD IX _L , A*	DDCB d 30	LD B, SLL (IX + d)*	DDCB d 83	LD E, RES 0, (IX + d)*
DD70 d	LD (IX + d), B	DDCB d 31	LD C, SLL (IX + d)*	DDCB d 84	LD H, RES 0, (IX + d)*
DD71 d	LD (IX + d), C	DDCB d 32	LD D, SLL (IX + d)*	DDCB d 85	LD L, RES 0, (IX + d)*
DD72 d	LD (IX + d), D	DDCB d 33	LD E, SLL (IX + d)*	DDCB d 86	RES 0, (IX + d)
DD73 d	LD (IX + d), E	DDCB d 34	LD H, SLL (IX + d)*	DDCB d 87	LD A, RES 0, (IX + d)*
DD74 d	LD (IX + d), H	DDCB d 35	LD L, SLL (IX + d)*	DDCB d 88	LD B, RES 1, (IX + d)*
DD75 d	LD (IX + d), L	DDCB d 36	SLL (IX + d)*	DDCB d 89	LD C, RES 1, (IX + d)*
DD77 d	LD (IX + d), A	DDCB d 37	LD A, SLL (IX + d)*	DDCB d 8A	LD D, RES 1, (IX + d)*
DD7C	LD A, IX _H *	DDCB d 38	LD B, SRL (IX + d)*	DDCB d 8B	LD E, RES 1, (IX + d)*
DD7D	LD A, IX _L *	DDCB d 39	LD C, SRL (IX + d)*	DDCB d 8C	LD H, RES 1, (IX + d)*
DD7E d	LD A, (IX + d)	DDCB d 3A	LD D, SRL (IX + d)*	DDCB d 8D	LD L, RES 1, (IX + d)*
DD84	ADD A, IX _H *	DDCB d 3B	LD E, SRL (IX + d)*	DDCB d 8E	RES 1, (IX + d)
DD85	ADD A, IX _L *	DDCB d 3C	LD H, SRL (IX + d)*	DDCB d 8F	LD A, RES 1, (IX + d)*
DD86 d	ADD A, (IX + d)	DDCB d 3D	LD L, SRL (IX + d)*	DDCB d 90	LD B, RES 2, (IX + d)*
DD8C	ADC A, IX _H *	DDCB d 3E	SRL (IX + d)	DDCB d 91	LD C, RES 2, (IX + d)*
DD8D	ADC A, IX _L *	DDCB d 3F	LD A, SRL (IX + d)*	DDCB d 92	LD D, RES 2, (IX + d)*
DD8E d	ADC A, (IX + d)	DDCB d 40	BIT 0, (IX + d)*	DDCB d 93	LD E, RES 2, (IX + d)*
DD94	SUB IX _H *	DDCB d 41	BIT 0, (IX + d)	DDCB d 94	LD H, RES 2, (IX + d)*
DD95	SUB IX _L *	DDCB d 42	BIT 0, (IX + d)*	DDCB d 95	LD L, RES 2, (IX + d)*
DD96 d	SUB (IX + d)	DDCB d 43	BIT 0, (IX + d)*	DDCB d 96	RES 2, (IX + d)
DD9C	SBC A, IX _H *	DDCB d 44	BIT 0, (IX + d)*	DDCB d 97	LD A, RES 2, (IX + d)*

DDCB d 98	LD B, RES 3, (IX + d)*	DDCB d EB	LD E, SET 5, (IX + d)*	ED6A	ADC HL, HL
DDCB d 99	LD C, RES 3, (IX + d)*	DDCB d EC	LD H, SET 5, (IX + d)*	ED6B n n	LD HL, (nn)
DDCB d 9A	LD D, RES 3, (IX + d)*	DDCB d ED	LD L, SET 5, (IX + d)*	ED6C	NEG*
DDCB d 9B	LD E, RES 3, (IX + d)*	DDCB d EE	SET 5, (IX + d)	ED6D	RETN*
DDCB d 9C	LD H, RES 3, (IX + d)*	DDCB d EF	LD A, SET 5, (IX + d)*	ED6E	IM 0/1*
DDCB d 9D	LD L, RES 3, (IX + d)*	DDCB d F0	LD B, SET 6, (IX + d)*	ED6F	RLD
DDCB d 9E	RES 3, (IX + d)	DDCB d F1	LD C, SET 6, (IX + d)*	ED70	IN (C)* / IN F, (C)*
DDCB d 9F	LD A, RES 3, (IX + d)*	DDCB d F2	LD D, SET 6, (IX + d)*	ED71	OUT (C), 0*
DDCB d A0	LD B, RES 4, (IX + d)*	DDCB d F3	LD E, SET 6, (IX + d)*	ED72	SBC HL, SP
DDCB d A1	LD C, RES 4, (IX + d)*	DDCB d F4	LD H, SET 6, (IX + d)*	ED73 n n	LD (nn), SP
DDCB d A2	LD D, RES 4, (IX + d)*	DDCB d F5	LD L, SET 6, (IX + d)*	ED74	NEG*
DDCB d A3	LD E, RES 4, (IX + d)*	DDCB d F6	SET 6, (IX + d)	ED75	RETN*
DDCB d A4	LD H, RES 4, (IX + d)*	DDCB d F7	LD A, SET 6, (IX + d)*	ED76	IM 1*
DDCB d A5	LD L, RES 4, (IX + d)*	DDCB d F8	LD B, SET 7, (IX + d)*	ED78	IN A, (C)
DDCB d A6	RES 4, (IX + d)	DDCB d F9	LD C, SET 7, (IX + d)*	ED79	OUT (C), A
DDCB d A7	LD A, RES 4, (IX + d)*	DDCB d FA	LD D, SET 7, (IX + d)*	ED7A	ADC HL, SP
DDCB d A8	LD B, RES 5, (IX + d)*	DDCB d FB	LD E, SET 7, (IX + d)*	ED7B n n	LD SP, (nn)
DDCB d A9	LD C, RES 5, (IX + d)*	DDCB d FC	LD H, SET 7, (IX + d)*	ED7C	NEG*
DDCB d AA	LD D, RES 5, (IX + d)*	DDCB d FD	LD L, SET 7, (IX + d)*	ED7D	RETN*
DDCB d AB	LD E, RES 5, (IX + d)*	DDCB d FE	SET 7, (IX + d)	ED7E	IM 2*
DDCB d AC	LD H, RES 5, (IX + d)*	DDCB d FF	LD A, SET 7, (IX + d)*	EDA0	LDI
DDCB d AD	LD L, RES 5, (IX + d)*	DDE1	POP IX	EDA1	CPI
DDCB d AE	RES 5, (IX + d)	DDE3	EX (SP), IX	EDA2	INI
DDCB d AF	LD A, RES 5, (IX + d)*	DDE5	PUSH IX	EDA3	OUTI
DDCB d B0	LD B, RES 6, (IX + d)*	DDE9	JP (IX)	EDA8	LDD
DDCB d B1	LD C, RES 6, (IX + d)*	DDF9	LD SP, IX	EDA9	CPD
DDCB d B2	LD D, RES 6, (IX + d)*	DE n	SBC A, n	EDAA	IND
DDCB d B3	LD E, RES 6, (IX + d)*	DF	RST 18h	EDAB	OUTD
DDCB d B4	LD H, RES 6, (IX + d)*	E0	RET PO	EDB0	LDIR
DDCB d B5	LD L, RES 6, (IX + d)*	E1	POP HL	EDB1	CPIR
DDCB d B6	RES 6, (IX + d)	E2 n n	JP PO, nn	EDB2	INIR
DDCB d B7	LD A, RES 6, (IX + d)*	E3	EX (SP), HL	EDB3	OTIR
DDCB d B8	LD B, RES 7, (IX + d)*	E4 n n	CALL PO, nn	EDB8	LDDR
DDCB d B9	LD C, RES 7, (IX + d)*	E5	PUSH HL	EDB9	CPDR
DDCB d BA	LD D, RES 7, (IX + d)*	E6 n	AND n	EDBA	INDR
DDCB d BB	LD E, RES 7, (IX + d)*	E7	RST 20h	EDBB	OTDR
DDCB d BC	LD H, RES 7, (IX + d)*	E8	RET PE	EE n	XOR n
DDCB d BD	LD L, RES 7, (IX + d)*	E9	JP (HL)	EF	RST 28h
DDCB d BE	RES 7, (IX + d)	EA n n	JP PE, (nn)	F0	RET P
DDCB d BF	LD A, RES 7, (IX + d)*	EB	EX DE, HL	F1	POP AF
DDCB d C0	LD B, SET 0, (IX + d)*	EC n n	CALL PE, nn	F2 n n	JP P, nn
DDCB d C1	LD C, SET 0, (IX + d)*	ED40	IN B, (C)	F3	DI
DDCB d C2	LD D, SET 0, (IX + d)*	ED41	OUT (C), B	F4 n n	CALL P, nn
DDCB d C3	LD E, SET 0, (IX + d)*	ED42	SBC HL, BC	F5	PUSH AF
DDCB d C4	LD H, SET 0, (IX + d)*	ED43 n n	LD (nn), BC	F6 n	OR n
DDCB d C5	LD L, SET 0, (IX + d)*	ED44	NEG	F7	RST 30h
DDCB d C6	SET 0, (IX + d)	ED45	RETN	F8	RET M
DDCB d C7	LD A, SET 0, (IX + d)*	ED46	IM 0	F9	LD SP, HL
DDCB d C8	LD B, SET 1, (IX + d)*	ED47	LD I, A	FA n n	JP M, nn
DDCB d C9	LD C, SET 1, (IX + d)*	ED48	IN C, (C)	FB	EI
DDCB d CA	LD D, SET 1, (IX + d)*	ED49	OUT (C), C	FC n n	CALL M, nn
DDCB d CB	LD E, SET 1, (IX + d)*	ED4A	ADC HL, BC	FD09	ADD IY, BC
DDCB d CC	LD H, SET 1, (IX + d)*	ED4B n n	LD BC, (nn)	FD19	ADD IY, DE
DDCB d CD	LD L, SET 1, (IX + d)*	ED4C	NEG*	FD21 n n	LD IY, nn
DDCB d CE	SET 1, (IX + d)	ED4D	RETI	FD22 n n	LD (nn), IY
DDCB d CF	LD A, SET 1, (IX + d)*	ED4E	IM 0/1*	FD23	INC IY
DDCB d D0	LD B, SET 2, (IX + d)*	ED4F	LD R, A	FD24	INC IY _H *
DDCB d D1	LD C, SET 2, (IX + d)*	ED50	IN D, (C)	FD25	DEC IY _H *
DDCB d D2	LD D, SET 2, (IX + d)*	ED51	OUT (C), D	FD26 n	LD IY _H , n*
DDCB d D3	LD E, SET 2, (IX + d)*	ED52	SBC HL, DE	FD29	ADD IY, IY
DDCB d D4	LD H, SET 2, (IX + d)*	ED53 n n	LD (nn), DE	FD2A n n	LD IY, (nn)
DDCB d D5	LD L, SET 2, (IX + d)*	ED54	NEG*	FD2B	DEC IY
DDCB d D6	SET 2, (IX + d)	ED55	RETN*	FD2C	INC IY _L *
DDCB d D7	LD A, SET 2, (IX + d)*	ED56	IM 1	FD2D	DEC IY _L *
DDCB d D8	LD B, SET 3, (IX + d)*	ED57	LD A, I	FD2E n	LD IY _L , n*
DDCB d D9	LD C, SET 3, (IX + d)*	ED58	IN E, (C)	FD34 d	INC (IY + d)
DDCB d DA	LD D, SET 3, (IX + d)*	ED59	OUT (C), E	FD35 d	DEC (IY + d)
DDCB d DB	LD E, SET 3, (IX + d)*	ED5A	ADC HL, DE	FD36 d n	LD (IY + d), n
DDCB d DC	LD H, SET 3, (IX + d)*	ED5B n n	LD DE, (nn)	FD39	ADD IY, SP
DDCB d DD	LD L, SET 3, (IX + d)*	ED5C	NEG*	FD44	LD B, IY _H *
DDCB d DE	SET 3, (IX + d)	ED5D	RETN*	FD45	LD B, IY _L *
DDCB d DF	LD A, SET 3, (IX + d)*	ED5E	IM 2	FD46 d	LD B, (IY + d)
DDCB d E0	LD B, SET 4, (IX + d)*	ED5F	LD A, R	FD4C	LD C, IY _H *
DDCB d E1	LD C, SET 4, (IX + d)*	ED60	IN H, (C)	FD4D	LD C, IY _L *
DDCB d E2	LD D, SET 4, (IX + d)*	ED61	OUT (C), H	FD4E d	LD C, (IY + d)
DDCB d E3	LD E, SET 4, (IX + d)*	ED62	SBC HL, HL	FD54	LD D, IY _H *
DDCB d E4	LD H, SET 4, (IX + d)*	ED63 n n	LD (nn), HL	FD55	LD D, IY _L *
DDCB d E5	LD L, SET 4, (IX + d)*	ED64	NEG*	FD56 d	LD D, (IY + d)
DDCB d E6	SET 4, (IX + d)	ED65	RETN*	FD5C	LD E, IY _H *
DDCB d E7	LD A, SET 4, (IX + d)*	ED66	IM 0*	FD5D	LD E, IY _L *
DDCB d E8	LD B, SET 5, (IX + d)*	ED67	RRD	FD5E d	LD E, (IY + d)
DDCB d E9	LD C, SET 5, (IX + d)*	ED68	IN L, (C)	FD60	LD IY _H , B*
DDCB d EA	LD D, SET 5, (IX + d)*	ED69	OUT (C), L	FD61	LD IY _H , C*

FD62	LD Y_H, D^*	FDCB d 23	LD E, SLA $(Y + d)^*$	FDCB d 76	BIT 6, $(Y + d)$
FD63	LD Y_H, E^*	FDCB d 24	LD H, SLA $(Y + d)^*$	FDCB d 77	BIT 6, $(Y + d)^*$
FD64	LD Y_H, Y_H^*	FDCB d 25	LD L, SLA $(Y + d)^*$	FDCB d 78	BIT 7, $(Y + d)^*$
FD65	LD Y_H, Y_L^*	FDCB d 26	SLA $(Y + d)$	FDCB d 79	BIT 7, $(Y + d)^*$
FD66 d	LD H, $(Y + d)$	FDCB d 27	LD A, SLA $(Y + d)^*$	FDCB d 7A	BIT 7, $(Y + d)^*$
FD67	LD Y_H, A^*	FDCB d 28	LD B, SRA $(Y + d)^*$	FDCB d 7B	BIT 7, $(Y + d)^*$
FD68	LD Y_L, B^*	FDCB d 29	LD C, SRA $(Y + d)^*$	FDCB d 7C	BIT 7, $(Y + d)^*$
FD69	LD Y_L, C^*	FDCB d 2A	LD D, SRA $(Y + d)^*$	FDCB d 7D	BIT 7, $(Y + d)^*$
FD6A	LD Y_L, D^*	FDCB d 2B	LD E, SRA $(Y + d)^*$	FDCB d 7E	BIT 7, $(Y + d)$
FD6B	LD Y_L, E^*	FDCB d 2C	LD H, SRA $(Y + d)^*$	FDCB d 7F	BIT 7, $(Y + d)^*$
FD6C	LD Y_L, Y_H^*	FDCB d 2D	LD L, SRA $(Y + d)^*$	FDCB d 80	LD B, RES 0, $(Y + d)^*$
FD6D	LD Y_L, Y_L^*	FDCB d 2E	SRA $(Y + d)$	FDCB d 81	LD C, RES 0, $(Y + d)^*$
FD6E d	LD L, $(Y + d)$	FDCB d 2F	LD A, SRA $(Y + d)^*$	FDCB d 82	LD D, RES 0, $(Y + d)^*$
FD6F	LD Y_L, A^*	FDCB d 30	LD B, SLL $(Y + d)^*$	FDCB d 83	LD E, RES 0, $(Y + d)^*$
FD70 d	LD $(Y + d), B$	FDCB d 31	LD C, SLL $(Y + d)^*$	FDCB d 84	LD H, RES 0, $(Y + d)^*$
FD71 d	LD $(Y + d), C$	FDCB d 32	LD D, SLL $(Y + d)^*$	FDCB d 85	LD L, RES 0, $(Y + d)^*$
FD72 d	LD $(Y + d), D$	FDCB d 33	LD E, SLL $(Y + d)^*$	FDCB d 86	RES 0, $(Y + d)$
FD73 d	LD $(Y + d), E$	FDCB d 34	LD H, SLL $(Y + d)^*$	FDCB d 87	LD A, RES 0, $(Y + d)^*$
FD74 d	LD $(Y + d), H$	FDCB d 35	LD L, SLL $(Y + d)^*$	FDCB d 88	LD B, RES 1, $(Y + d)^*$
FD75 d	LD $(Y + d), L$	FDCB d 36	SLL $(Y + d)^*$	FDCB d 89	LD C, RES 1, $(Y + d)^*$
FD77 d	LD $(Y + d), A$	FDCB d 37	LD A, SLL $(Y + d)^*$	FDCB d 8A	LD D, RES 1, $(Y + d)^*$
FD7C	LD A, Y_H^*	FDCB d 38	LD B, SRL $(Y + d)^*$	FDCB d 8B	LD E, RES 1, $(Y + d)^*$
FD7D	LD A, Y_L^*	FDCB d 39	LD C, SRL $(Y + d)^*$	FDCB d 8C	LD H, RES 1, $(Y + d)^*$
FD7E d	LD A, $(Y + d)$	FDCB d 3A	LD D, SRL $(Y + d)^*$	FDCB d 8D	LD L, RES 1, $(Y + d)^*$
FD84	ADD A, Y_H^*	FDCB d 3B	LD E, SRL $(Y + d)^*$	FDCB d 8E	RES 1, $(Y + d)$
FD85	ADD A, Y_L^*	FDCB d 3C	LD H, SRL $(Y + d)^*$	FDCB d 8F	LD A, RES 1, $(Y + d)^*$
FD86 d	ADD A, $(Y + d)$	FDCB d 3D	LD L, SRL $(Y + d)^*$	FDCB d 90	LD B, RES 2, $(Y + d)^*$
FD8C	ADC A, Y_H^*	FDCB d 3E	SRL $(Y + d)$	FDCB d 91	LD C, RES 2, $(Y + d)^*$
FD8D	ADC A, Y_L^*	FDCB d 3F	LD A, SRL $(Y + d)^*$	FDCB d 92	LD D, RES 2, $(Y + d)^*$
FD8E d	ADC A, $(Y + d)$	FDCB d 40	BIT 0, $(Y + d)^*$	FDCB d 93	LD E, RES 2, $(Y + d)^*$
FD94	SUB Y_H^*	FDCB d 41	BIT 0, $(Y + d)^*$	FDCB d 94	LD H, RES 2, $(Y + d)^*$
FD95	SUB Y_L^*	FDCB d 42	BIT 0, $(Y + d)^*$	FDCB d 95	LD L, RES 2, $(Y + d)^*$
FD96 d	SUB $(Y + d)$	FDCB d 43	BIT 0, $(Y + d)^*$	FDCB d 96	RES 2, $(Y + d)$
FD9C	SBC A, Y_H^*	FDCB d 44	BIT 0, $(Y + d)^*$	FDCB d 97	LD A, RES 2, $(Y + d)^*$
FD9D	SBC A, Y_L^*	FDCB d 45	BIT 0, $(Y + d)^*$	FDCB d 98	LD B, RES 3, $(Y + d)^*$
FD9E d	SBC A, $(Y + d)$	FDCB d 46	BIT 0, $(Y + d)$	FDCB d 99	LD C, RES 3, $(Y + d)^*$
FDA4	AND Y_H^*	FDCB d 47	BIT 0, $(Y + d)^*$	FDCB d 9A	LD D, RES 3, $(Y + d)^*$
FDA5	AND Y_L^*	FDCB d 48	BIT 1, $(Y + d)^*$	FDCB d 9B	LD E, RES 3, $(Y + d)^*$
FDA6 d	AND $(Y + d)$	FDCB d 49	BIT 1, $(Y + d)^*$	FDCB d 9C	LD H, RES 3, $(Y + d)^*$
FDAC	XOR Y_H^*	FDCB d 4A	BIT 1, $(Y + d)^*$	FDCB d 9D	LD L, RES 3, $(Y + d)^*$
FDAD	XOR Y_L^*	FDCB d 4B	BIT 1, $(Y + d)^*$	FDCB d 9E	RES 3, $(Y + d)$
FDAE d	XOR $(Y + d)$	FDCB d 4C	BIT 1, $(Y + d)^*$	FDCB d 9F	LD A, RES 3, $(Y + d)^*$
FDB4	OR Y_H^*	FDCB d 4D	BIT 1, $(Y + d)^*$	FDCB d A0	LD B, RES 4, $(Y + d)^*$
FDB5	OR Y_L^*	FDCB d 4E	BIT 1, $(Y + d)$	FDCB d A1	LD C, RES 4, $(Y + d)^*$
FDB6 d	OR $(Y + d)$	FDCB d 4F	BIT 1, $(Y + d)^*$	FDCB d A2	LD D, RES 4, $(Y + d)^*$
FDBC	CP Y_H^*	FDCB d 50	BIT 2, $(Y + d)^*$	FDCB d A3	LD E, RES 4, $(Y + d)^*$
FDBD	CP Y_L^*	FDCB d 51	BIT 2, $(Y + d)^*$	FDCB d A4	LD H, RES 4, $(Y + d)^*$
FDBE d	CP $(Y + d)$	FDCB d 52	BIT 2, $(Y + d)^*$	FDCB d A5	LD L, RES 4, $(Y + d)^*$
FDCB d 00	LD B, RLC $(Y + d)^*$	FDCB d 53	BIT 2, $(Y + d)^*$	FDCB d A6	RES 4, $(Y + d)$
FDCB d 01	LD C, RLC $(Y + d)^*$	FDCB d 54	BIT 2, $(Y + d)^*$	FDCB d A7	LD A, RES 4, $(Y + d)^*$
FDCB d 02	LD D, RLC $(Y + d)^*$	FDCB d 55	BIT 2, $(Y + d)^*$	FDCB d A8	LD B, RES 5, $(Y + d)^*$
FDCB d 03	LD E, RLC $(Y + d)^*$	FDCB d 56	BIT 2, $(Y + d)$	FDCB d A9	LD C, RES 5, $(Y + d)^*$
FDCB d 04	LD H, RLC $(Y + d)^*$	FDCB d 57	BIT 2, $(Y + d)^*$	FDCB d AA	LD D, RES 5, $(Y + d)^*$
FDCB d 05	LD L, RLC $(Y + d)^*$	FDCB d 58	BIT 3, $(Y + d)^*$	FDCB d AB	LD E, RES 5, $(Y + d)^*$
FDCB d 06	RLC $(Y + d)$	FDCB d 59	BIT 3, $(Y + d)^*$	FDCB d AC	LD H, RES 5, $(Y + d)^*$
FDCB d 07	LD A, RLC $(Y + d)^*$	FDCB d 5A	BIT 3, $(Y + d)^*$	FDCB d AD	LD L, RES 5, $(Y + d)^*$
FDCB d 08	LD B, RRC $(Y + d)^*$	FDCB d 5B	BIT 3, $(Y + d)^*$	FDCB d AE	RES 5, $(Y + d)$
FDCB d 09	LD C, RRC $(Y + d)^*$	FDCB d 5C	BIT 3, $(Y + d)^*$	FDCB d AF	LD A, RES 5, $(Y + d)^*$
FDCB d 0A	LD D, RRC $(Y + d)^*$	FDCB d 5D	BIT 3, $(Y + d)^*$	FDCB d B0	LD B, RES 6, $(Y + d)^*$
FDCB d 0B	LD E, RRC $(Y + d)^*$	FDCB d 5E	BIT 3, $(Y + d)$	FDCB d B1	LD C, RES 6, $(Y + d)^*$
FDCB d 0C	LD H, RRC $(Y + d)^*$	FDCB d 5F	BIT 3, $(Y + d)^*$	FDCB d B2	LD D, RES 6, $(Y + d)^*$
FDCB d 0D	LD L, RRC $(Y + d)^*$	FDCB d 60	BIT 4, $(Y + d)^*$	FDCB d B3	LD E, RES 6, $(Y + d)^*$
FDCB d 0E	RRC $(Y + d)$	FDCB d 61	BIT 4, $(Y + d)^*$	FDCB d B4	LD H, RES 6, $(Y + d)^*$
FDCB d 0F	LD A, RRC $(Y + d)^*$	FDCB d 62	BIT 4, $(Y + d)^*$	FDCB d B5	LD L, RES 6, $(Y + d)^*$
FDCB d 10	LD B, RL $(Y + d)^*$	FDCB d 63	BIT 4, $(Y + d)^*$	FDCB d B6	RES 6, $(Y + d)$
FDCB d 11	LD C, RL $(Y + d)^*$	FDCB d 64	BIT 4, $(Y + d)^*$	FDCB d B7	LD A, RES 6, $(Y + d)^*$
FDCB d 12	LD D, RL $(Y + d)^*$	FDCB d 65	BIT 4, $(Y + d)^*$	FDCB d B8	LD B, RES 7, $(Y + d)^*$
FDCB d 13	LD E, RL $(Y + d)^*$	FDCB d 66	BIT 4, $(Y + d)$	FDCB d B9	LD C, RES 7, $(Y + d)^*$
FDCB d 14	LD H, RL $(Y + d)^*$	FDCB d 67	BIT 4, $(Y + d)^*$	FDCB d BA	LD D, RES 7, $(Y + d)^*$
FDCB d 15	LD L, RL $(Y + d)^*$	FDCB d 68	BIT 5, $(Y + d)^*$	FDCB d BB	LD E, RES 7, $(Y + d)^*$
FDCB d 16	RL $(Y + d)$	FDCB d 69	BIT 5, $(Y + d)^*$	FDCB d BC	LD H, RES 7, $(Y + d)^*$
FDCB d 17	LD A, RL $(Y + d)^*$	FDCB d 6A	BIT 5, $(Y + d)^*$	FDCB d BD	LD L, RES 7, $(Y + d)^*$
FDCB d 18	LD B, RR $(Y + d)^*$	FDCB d 6B	BIT 5, $(Y + d)^*$	FDCB d BE	RES 7, $(Y + d)$
FDCB d 19	LD C, RR $(Y + d)^*$	FDCB d 6C	BIT 5, $(Y + d)^*$	FDCB d BF	LD A, RES 7, $(Y + d)^*$
FDCB d 1A	LD D, RR $(Y + d)^*$	FDCB d 6D	BIT 5, $(Y + d)^*$	FDCB d C0	LD B, SET 0, $(Y + d)^*$
FDCB d 1B	LD E, RR $(Y + d)^*$	FDCB d 6E	BIT 5, $(Y + d)$	FDCB d C1	LD C, SET 0, $(Y + d)^*$
FDCB d 1C	LD H, RR $(Y + d)^*$	FDCB d 6F	BIT 5, $(Y + d)^*$	FDCB d C2	LD D, SET 0, $(Y + d)^*$
FDCB d 1D	LD L, RR $(Y + d)^*$	FDCB d 70	BIT 6, $(Y + d)^*$	FDCB d C3	LD E, SET 0, $(Y + d)^*$
FDCB d 1E	RR $(Y + d)$	FDCB d 71	BIT 6, $(Y + d)^*$	FDCB d C4	LD H, SET 0, $(Y + d)^*$
FDCB d 1F	LD A, RR $(Y + d)^*$	FDCB d 72	BIT 6, $(Y + d)^*$	FDCB d C5	LD L, SET 0, $(Y + d)^*$
FDCB d 20	LD B, SLA $(Y + d)^*$	FDCB d 73	BIT 6, $(Y + d)^*$	FDCB d C6	SET 0, $(Y + d)$
FDCB d 21	LD C, SLA $(Y + d)^*$	FDCB d 74	BIT 6, $(Y + d)^*$	FDCB d C7	LD A, SET 0, $(Y + d)^*$
FDCB d 22	LD D, SLA $(Y + d)^*$	FDCB d 75	BIT 6, $(Y + d)^*$	FDCB d C8	LD B, SET 1, $(Y + d)^*$

FDCB d C9	LD C, SET 1, (IY + d)*
FDCB d CA	LD D, SET 1, (IY + d)*
FDCB d CB	LD E, SET 1, (IY + d)*
FDCB d CC	LD H, SET 1, (IY + d)*
FDCB d CD	LD L, SET 1, (IY + d)*
FDCB d CE	SET 1, (IY + d)
FDCB d CF	LD A, SET 1, (IY + d)*
FDCB d D0	LD B, SET 2, (IY + d)*
FDCB d D1	LD C, SET 2, (IY + d)*
FDCB d D2	LD D, SET 2, (IY + d)*
FDCB d D3	LD E, SET 2, (IY + d)*
FDCB d D4	LD H, SET 2, (IY + d)*
FDCB d D5	LD L, SET 2, (IY + d)*
FDCB d D6	SET 2, (IY + d)
FDCB d D7	LD A, SET 2, (IY + d)*
FDCB d D8	LD B, SET 3, (IY + d)*
FDCB d D9	LD C, SET 3, (IY + d)*
FDCB d DA	LD D, SET 3, (IY + d)*
FDCB d DB	LD E, SET 3, (IY + d)*
FDCB d DC	LD H, SET 3, (IY + d)*
FDCB d DD	LD L, SET 3, (IY + d)*
FDCB d DE	SET 3, (IY + d)
FDCB d DF	LD A, SET 3, (IY + d)*
FDCB d E0	LD B, SET 4, (IY + d)*
FDCB d E1	LD C, SET 4, (IY + d)*
FDCB d E2	LD D, SET 4, (IY + d)*
FDCB d E3	LD E, SET 4, (IY + d)*
FDCB d E4	LD H, SET 4, (IY + d)*
FDCB d E5	LD L, SET 4, (IY + d)*
FDCB d E6	SET 4, (IY + d)
FDCB d E7	LD A, SET 4, (IY + d)*
FDCB d E8	LD B, SET 5, (IY + d)*
FDCB d E9	LD C, SET 5, (IY + d)*
FDCB d EA	LD D, SET 5, (IY + d)*
FDCB d EB	LD E, SET 5, (IY + d)*
FDCB d EC	LD H, SET 5, (IY + d)*
FDCB d ED	LD L, SET 5, (IY + d)*
FDCB d EE	SET 5, (IY + d)
FDCB d EF	LD A, SET 5, (IY + d)*
FDCB d F0	LD B, SET 6, (IY + d)*
FDCB d F1	LD C, SET 6, (IY + d)*
FDCB d F2	LD D, SET 6, (IY + d)*
FDCB d F3	LD E, SET 6, (IY + d)*
FDCB d F4	LD H, SET 6, (IY + d)*
FDCB d F5	LD L, SET 6, (IY + d)*
FDCB d F6	SET 6, (IY + d)
FDCB d F7	LD A, SET 6, (IY + d)*
FDCB d F8	LD B, SET 7, (IY + d)*
FDCB d F9	LD C, SET 7, (IY + d)*
FDCB d FA	LD D, SET 7, (IY + d)*
FDCB d FB	LD E, SET 7, (IY + d)*
FDCB d FC	LD H, SET 7, (IY + d)*
FDCB d FD	LD L, SET 7, (IY + d)*
FDCB d FE	SET 7, (IY + d)
FDCB d FF	LD A, SET 7, (IY + d)*
FDE1	POP IY
FDE3	EX (SP), IY
FDE5	PUSH IY
FDE9	JP (IY)
FDF9	LD SP, IY
FE n	CP n
FF	RST 38h