Zilog Z80 CPU Specifications

by Sean Young (syoung@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 the are two flip-flops associated with interrupts: IFF_1 and IFF_2. DI resets both and EI sets both. If IFF_1 is set then mask-able interrupts are accepted. When an non mask-able interrupt occurs, the IFF_1 is reset, so disabling mask-able interrupts. When the CPU returns from a non mask-able interrupt (with RETN) the IFF_2 is copied into IFF_1 (and thus restored). Only IFF_2 can be read (with LD A, I and LD A, R IFF_2 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 $I \times 256$ + 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 $I \times 256$ + FFh).

Instructions Descriptions

8 bit Load Group

Mnemonic	Symbolic Operation	Flags 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 ? r'	•	•	•	•	•	•	•	•	01 r r'		1	1	4	r, r' Reg.
LD p, p'*	p ? p'	•	•	•	•	•	•	•	•	11 011 101 / 01 p p'	DD	2	2	8	000 B
LD q, q'*	q ? q'	•	•	•	•	•	•	•	•	11 111 101 / 01 q q'	FD	2	2	8	001 C / 010 D / 011 E
LD r, n	r ? n	•	•	•	•	•	•	•	•	00 r 110 / ? n ?		2	2	7	100 H / 101 L
LD p, n*	p ? n	•	•	•	•	•	•	•	•	11 011 101 / 00 p 110 / ? n ?	DD	3	3	11	111 A
LD q, n*	q ? n	•	•	•	•	•	•	•	•	11 111 101 / 00 q 110 / ? n ?	FD	3	3	11	p, p' Reg. / 000 B / 001 C / 010 D
LD r, (HL)	r ? (HL)	•	•	•	•	•	•	•	•	01 r 110		1	2	7	011 E
LD r, (IX + d)	r ? (IX + d)	•	•	•	•	•	•	•	•	11 011 101 / 01 r 110 / ? d ?	DD	3	5	19	100 IX$_H$ / 101 IX$_L$ / 111 A
LD r, (IY + d)	r ? (IY + d)	•	•	•	•	•	•	•	•	11 111 101 / 01 r 110 / ? d ?	FD	3	5	19	q, q' Reg. / 000 B
LD (HL), r	(HL) ? r	•	•	•	•	•	•	•	•	01 110 r		1	2	7	001 C
LD (IX + d), r	(IX + d) ? r	•	•	•	•	•	•	•	•	11 011 101 / 01 110 r / ? d ?	DD	3	5	19	010 D / 011 E / 100 IY$_H$
LD (IY + d), r	(IY + d) ? r	•	•	•	•	•	•	•	•	11 111 101 / 01 110 r / ? d ?	FD	3	5	19	101 IY$_L$ / 111 A
LD (HL), n	(HL) ? n	•	•	•	•	•	•	•	•	00 110 110 / ? n ?	36	2	3	10	
LD (IX + d), n	(IX + d) ? n	•	•	•	•	•	•	•	•	11 011 101 / 00 110 110 / ? d ? / ? n ?	DD 36	4	5	19	
LD (IY + d), n	(IY + d) ? n	•	•	•	•	•	•	•	•	11 111 101 / 00 110 110 / ? d ? / ? n ?	FD 36	4	5	19	
LD A, (BC)	A ? (BC)	•	•	•	•	•	•	•	•	00 001 010	0A	1	2	7	
LD A, (DE)	A ? (DE)	•	•	•	•	•	•	•	•	00 011 010	1A	1	2	7	
LD A, (nn)	A ? (nn)	•	•	•	•	•	•	•	•	00 111 010 / ? n ? / ? n ?	3A	3	4	13	
LD (BC), A	(BC) ? A	•	•	•	•	•	•	•	•	00 000 010	02	1	2	7	
LD (DE), A	(DE) ? A	•	•	•	•	•	•	•	•	00 010 010	12	1	2	7	
LD (nn), A	(nn) ? A	•	•	•	•	•	•	•	•	00 110 010 / ? n ? / ? n ?	32	3	4	13	
LD A, I	A ? I	?	?	?	0	?	IFF$_2$	0	•	11 101 101 / 01 010 111	ED 57	2	2	9	
LD A, R	A ? R	?	?	?	0	?	IFF$_2$	0	•	11 101 101 / 01 011 111	ED 5F	2	2	9	R is read after it is increased.
LD I, A	I ? A	•	•	•	•	•	•	•	•	11 101 101 / 01 000 111	ED 47	2	2	9	
LD R, A	R ? A	•	•	•	•	•	•	•	•	11 101 101 / 01 001 111	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,
? = flag is set according to the result of the operation, IFF$_2$ = the interrupt flip-flop 2 is copied.

16 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 dd, nn	dd ← nn	•	•	•	•	•	•	•	•	00 dd0 001 ← n → ← n →		3	3	10	dd Pair 00 BC 01 DE 02 HL 03 SP
LD IX, nn	IX ← nn	•	•	•	•	•	•	•	•	11 011 101 00 110 001 ← n → ← n →	DD 21	4	4	14	
LD IY, nn	IY ← nn	•	•	•	•	•	•	•	•	11 111 101 00 110 001 ← n → ← n →	FD 21	4	4	14	
LD HL, (nn)	L ← (nn) H ← (nn+1)	•	•	•	•	•	•	•	•	00 101 010 ← n → ← n →	2A	3	5	16	
LD dd, (nn)	dd_L ← (nn) dd_H ← (nn+1)	•	•	•	•	•	•	•	•	11 101 101 01 dd1 011 ← n → ← n →	ED	4	6	20	
LD IX, (nn)	IX_L ← (nn) IX_H ← (nn+1)	•	•	•	•	•	•	•	•	11 011 101 00 101 010 ← n → ← n →	DD 2A	4	6	20	
LD IY, (nn)	IY_L ← (nn) IY_H ← (nn+1)	•	•	•	•	•	•	•	•	11 111 101 00 101 010 ← n → ← n →	FD 2A	4	6	20	
LD (nn), HL	(nn) ← L (nn+1) ← H	•	•	•	•	•	•	•	•	00 100 010 ← n → ← n →	22	3	5	16	
LD (nn), dd	(nn) ← dd_L (nn+1) ← dd_H	•	•	•	•	•	•	•	•	11 101 101 01 dd0 011 ← n → ← n →	DD	4	6	20	
LD (nn), IX	(nn) ← IX_L (nn+1) ← IX_H	•	•	•	•	•	•	•	•	11 011 101 00 100 010 ← n → ← n →	DD 22	4	6	20	
LD (nn), IY	(nn) ← IY_L (nn+1) ← IY_H	•	•	•	•	•	•	•	•	11 111 101 00 100 010 ← n → ← n →	FD 22	4	6	20	
LD SP, HL	SP ← HL	•	•	•	•	•	•	•	•	11 111 001	F9	1	1	6	
LD SP, IX	SP ← IX	•	•	•	•	•	•	•	•	11 011 101 11 111 001	DD F9	2	2	10	
LD SP, IY	SP ← IY	•	•	•	•	•	•	•	•	11 111 101 11 111 001	FD F9	2	2	10	
PUSH qq	SP ← SP - 1 (SP) ← qq_H SP ← SP - 1 (SP) ← qq_L	•	•	•	•	•	•	•	•	11 qq0 101		1	3	11	qq Pair 00 BC 01 DE 10 HL 11 AF
PUSH IX	SP ← SP - 1 (SP) ← IX_H SP ← SP - 1 (SP) ← IX_L	•	•	•	•	•	•	•	•	11 011 101 11 100 101	DD E5	2	4	15	
PUSH IY	SP ← SP - 1 (SP) ← IY_H SP ← SP - 1 (SP) ← IY_L	•	•	•	•	•	•	•	•	11 111 101 11 100 101	FD E5	2	4	15	
POP qq	(SP) ← qq_L SP ← SP + 1 (SP) ← qq_H SP ← SP + 1	•	•	•	•	•	•	•	•	11 qq0 001		1	3	10	
POP IX	(SP) ← IX_L SP ← SP + 1 (SP) ← IX_H SP ← SP + 1	•	•	•	•	•	•	•	•	11 011 101 11 100 001	DD E1	2	4	14	
POP IY	(SP) ← IY_L SP ← SP + 1 (SP) ← IY_H SP ← SP + 1	•	•	•	•	•	•	•	•	11 111 101 11 100 001	FD E1	2	4	14	

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 ? = flag is set according to the result of the operation.

Exchange, Block Transfer and Search Groups

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
EX DE, HL	DE ? HL	•	•	•	•	•	•	•	•	11 101 011	EB	1	1	4	
EX AF, AF'	AF ? AF'	•	•	•	•	•	•	•	•	00 001 000	08	1	1	4	
EXX	BC ? BC' DE ? DE' HL ? HL'	•	•	•	•	•	•	•	•	11 011 001	D9	1	1	4	
EX (SP), HL	(SP+1) ? H (SP) ? L	•	•	•	•	•	•	•	•	11 100 011	E3	1	5	19	
EX (SP), IX	(SP+1) ? IX$_H$ (SP) ? IX$_L$	•	•	•	•	•	•	•	•	11 011 101 11 100 011	DD E3	2	6	23	
EX (SP), IY	(SP+1) ? IY$_H$ (SP) ? IY$_L$	•	•	•	•	•	•	•	•	11 111 101 11 100 011	FD E3	2	6	23	
LDI	(DE) ? (HL) DE ? DE + 1 HL ? HL + 1 BC ? BC - 1	•	•	?[1]	0	?[2]	?[3]	0	•	11 101 101 10 100 000	ED A0	2	4	16	
LDIR	(DE) ? (HL) DE ? DE + 1 HL ? HL + 1 BC ? BC - 1 repeat until: BC = 0	•	•	?[1]	0	?[2]	0	0	•	11 101 101 10 110 000	ED B0	2 2	5 4	21 16	if BC ? 0 if BC = 0
LDD	(DE) ? (HL) DE ? DE - 1 HL ? HL - 1 BC ? BC - 1	•	•	?[1]	0	?[2]	?[3]	0	•	11 101 101 10 101 000	ED A8	2	4	16	
LDDR	(DE) ? (HL) DE ? DE - 1 HL ? HL - 1 BC ? BC - 1 repeat until: BC = 0	•	•	?[1]	0	?[2]	0	0	•	11 101 101 10 111 000	ED B8	2 2	5 4	21 16	if BC ? 0 if BC = 0
CPI	A - (HL) HL ? HL + 1 BC ? BC -1	?[4]	?[4]	?[5]	?[4]	?[6]	?[3]	1	•	11 101 101 10 100 001	ED A1	2	4	16	
CPIR	A - (HL) HL ? HL + 1 BC ? BC -1 Repeat until: A = (HL) or BC = 0	?[4]	?[4]	?[5]	?[4]	?[6]	?[3]	1	•	11 101 101 10 110 001	ED B1	2 2	5 4	21 16	if BC ? 0 and A ? (HL). if BC = 0 or A = (HL)
CPD	A - (HL) HL ? HL - 1 BC ? BC -1	?[4]	?[4]	?[5]	?[4]	?[6]	?[3]	1	•	11 101 101 10 101 001	ED A9	2	4	16	
CPDR	A - (HL) HL ? HL - 1 BC ? BC -1 Repeat until: A = (HL) or BC = 0	?[4]	?[4]	?[5]	?[4]	?[6]	?[3]	1	•	11 101 101 10 111 001	ED B9	2 2	5 4	21 16	if BC ? 0 and A ? (HL). if BC = 0 or A = (HL)

Notes:

[1] F5 is a copy of bit 1 of A + last transferred byte, thus (A + (HL))$_1$

[2] F3 is a copy of bit 3 of A + last transferred byte, thus (A + (HL))$_3$

[3] P/V flag is 0 if the result of BC- 1 = 0, otherwise P/V = 1.

[4] These flags are set as in CP (HL)

[5] F5 is copy of bit 1 of A- last compared address- H, thus (A - (HL) - H)$_1$. H is as in F after the comparison.

[6] F3 is copy of bit 3 of A- last compared address- H, thus (A - (HL) - H)$_3$. H is as in F after the comparison.

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

8 bit Arithmetic and Logical 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
ADD A, r	A ? A + r	?	?	?	?	?	V	0	?	10 000 r		1	1	4	r Reg. p Reg.
ADD A, p*	A ? A + p	?	?	?	?	?	V	0	?	11 011 101 / 10 000 p	DD	2	2	8	000 B 000 B
															001 C 001 C
ADD A, q*	A ? A + q	?	?	?	?	?	V	0	?	11 111 101 / 10 000 q	FD	2	2	8	010 D 010 D
															011 E 011 E
ADD A, n	A ? A + n	?	?	?	?	?	V	0	?	11 000 110 / ? n ?		2	2	8	100 H 100 IX_H
															101 L 101 IX_H
ADD A, (HL)	A ? A + (HL)	?	?	?	?	?	V	0	?	10 000 110		1	2	7	111 A 111 A
ADD A, (IX + d)	A ? A + (IX + d)	?	?	?	?	?	V	0	?	11 011 101 / 10 000 110 / ? d ?	DD	3	5	19	
ADD A, (IY + d)	A ? A + (IY + d)	?	?	?	?	?	V	0	?	11 111 101 / 10 000 110 / ? d ?	FD	3	5	19	
ADC A, s	A ? A + s + CY	?	?	?	?	?	V	0	?	001					s is any of r, n, (HL),
SUB A, s	A ? A - s	?	?	?	?	?	V	1	?	010					(IX+d), (IY+d), p, q
SBC A, s	A ? A - s - CY	?	?	?	?	?	V	1	?	011					as shown for the ADD
AND s	A ? A AND s	?	?	?	1	?	P	0	0	100					instruction. The
OR s	A ? A OR s	?	?	?	0	?	P	0	0	110					underlined bits replace
XOR s	A ? A XOR s	?	?	?	0	?	P	0	0	101					the underlined bits in
CP s	A - s	?	?	?[1]	?	?[1]	V	1	?	111					the ADD set.
INC r	r ? r + 1	?	?	?	?	?	V	0	•	00 r 100		1	1	4	
INC p*	p ? p + 1	?	?	?	?	?	V	0	•	11 011 101 / 00 p 100	DD	2	2	8	q Reg.
															000 B
INC q*	q ? q + 1	?	?	?	?	?	V	0	•	11 111 101 / 00 q 100	FD	2	2	8	001 C
															010 D
INC (HL)	(HL) ? (HL) + 1	?	?	?	?	?	V	0	•	00 110 100		1	3	11	011 E
INC (IX + d)	(IX + d) ? (IX + d) + 1	?	?	?	?	?	V	0	•	11 011 101 / 00 110 100 / ? d ?	DD	3	6	23	100 IY_H
															101 IY_L
															111 A
INC (IY + d)	(IY + d) ? (IY + d) + 1	?	?	?	?	?	V	0	•	11 111 101 / 00 110 100 / ? d ?	FD	3	6	23	
DEC m	m ? m - 1	?	?	?	?	?	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 (As).

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 flipflop.

* means unofficial instruction.

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

16 bit Arithmetic 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
ADD HL, ss	HL ? HL + ss	•	•	?[2]	?[2]	?[2]	•	0	?[1]	00 ss1 001		1	3	11	ss Reg.
ADC HL, ss	HL ? HL + ss + CY	?[1]	?[1]	?[2]	?[2]	?[2]	V[1]	0	?[1]	11 101 101 / 01 ss1 010	ED	2	4	15	00 BC / 01 DE
SBC HL, ss	HL ? HL - ss - CY	?[1]	?[1]	?[2]	?[2]	?[2]	V[1]	1	?[1]	11 101 101 / 01 ss0 010	ED	2	4	15	10 HL / 11 SP
ADD IX, pp	IX ? IX + pp	•	•	?[2]	?[2]	?[2]	•	0	?[1]	11 011 101 / 00 pp1 001	DD	2	4	15	
ADD IY, rr	IY ? IY + rr	•	•	?[2]	?[2]	?[2]	•	0	?[1]	11 111 101 / 00 rr1 001	FD	2	4	15	pp Reg. / 00 BC / 01 DE
INC ss	ss ? ss + 1	•	•	•	•	•	•	•	•	00 ss0 011		1	1	6	10 IX
INC IX	IX ? IX + 1	•	•	•	•	•	•	•	•	11 011 101 / 00 100 011	DD 23	2	2	10	11 SP
INC IY	IY ? IY + 1	•	•	•	•	•	•	•	•	11 111 101 / 00 100 011	FD 23	2	2	10	rr Reg. / 00 BC
DEC ss	ss ? ss - 1	•	•	•	•	•	•	•	•	00 ss1 011		1	1	6	01 DE
DEC IX	IX ? IX - 1	•	•	•	•	•	•	•	•	11 011 101 / 00 101 011	DD 2B	2	2	10	10 IY
DEC IY	IY ? IY - 1	•	•	•	•	•	•	•	•	11 111 101 / 00 101 011	FD 2B	2	2	10	11 SP

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 flipflop.

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

General Purpose Arithmetic and CPU Control Groups

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
DAA	Converts A into packed BCD following add or subtract with BCD operands.	?	?	?	?	?	P	•	?	00 100 111	27	1	1	4	
CPL	A ? $\overline{A}$	•	•	?[1]	1	?[1]	•	1	•	00 101 111	2F	1	1	4	One's complement.
NEG[4]	A ? 0 - A	?	?	?	?	?	V	1	?	11 101 101 / 01 000 100	ED 44	2	2	8	Two's complement.
CCF	CY ? $\overline{CY}$	•	•	?[1]	?[2]	?[1]	•	0	?	00 111 111	3F	1	1	4	Complement carry flag.
SCF	CY ? 1	•	•	?[1]	0	?[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 / IFF$_2$? 0	•	•	•	•	•	•	•	•	11 110 011	F3	1	1	4	
EI[3]	IFF$_1$? 1 / IFF$_2$? 1	•	•	•	•	•	•	•	•	11 111 011	FB	1	1	4	
IM 0[4]	Set interrupt mode 0	•	•	•	•	•	•	•	•	11 101 101 / 01 000 110	ED 46	2	2	8	
IM 1[4]	Set interrupt mode 1	•	•	•	•	•	•	•	•	11 101 101 / 01 010 110	ED 56	2	2	8	
IM 2[4]	Set interrupt mode 2	•	•	•	•	•	•	•	•	11 101 101 / 01 011 110	ED 5E	2	2	8	

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 flipflop.

Flag Notation: • = flag is not affected, 0 = flag is reset, 1 = flag is set, ? = 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	CY←[7←0]	•	•	?	0	?	•	0	?	00 000 111	07	1	1	4	
RLA	[CY←[7←0]	•	•	?	0	?	•	0	?	00 010 111	17	1	1	4	
RRCA	[7→0]→CY	•	•	?	0	?	•	0	?	00 001 111	0F	1	1	4	
RRA	[7→0]→CY]	•	•	?	0	?	•	0	?	00 011 111	1F	1	1	4	
RLC r	CY←[7←0]	?	?	?	0	?	P	0	?	11 001 011 00 000 r	CB	2	2	8	r Reg. 000 B
RLC (HL)	CY←[7←0]	?	?	?	0	?	P	0	?	11 001 011 00 000 110	CB	2	4	15	001 C 010 D
RLC (IX + d)	CY←[7←0]	?	?	?	0	?	P	0	?	11 011 101 11 001 011 ? d ? 00 000 110	DD CB	4	6	23	011 E 100 H 101 L 111 A
RLC (IY + d)	CY←[7←0]	?	?	?	0	?	P	0	?	11 111 101 11 001 011 ? d ? 00 000 110	FD CB	4	6	23	
LD r,RLC (IX + d)*	r ? (IX + d) RLC r (IX + d) ? r	?	?	?	0	?	P	0	?	11 011 101 11 001 011 ? d ? 00 000 r	DD CB	4	6	23	
LD r,RLC (IY + d)*	r ? (IY + d) RLC r (IY + d) ? r	?	?	?	0	?	P	0	?	11 111 101 11 001 011 ? d ? 00 000 r	FD CB	4	6	23	
RL m	[CY←[7←0]	?	?	?	0	?	P	0	?	010					Instruction format
RRC m	[7→0]→CY	?	?	?	0	?	P	0	?	001					and states are the
RR m	[7→0]→CY]	?	?	?	0	?	P	0	?	011					same as RLC.
SLA m	CY←[7←0]←0	?	?	?	0	?	P	0	?	100					Replace 000 with
SLL m*	CY←[7←0]←1	?	?	?	0	?	P	0	?	110					new number.
SRA m	[7→0]→CY	?	?	?	0	?	P	0	?	101					
SRL m	0→[7→0]→CY	?	?	?	0	?	P	0	?	111					
RLD	[0347] [0347] A (HL)	?	?	?	0	?	P	0	•	11 101 101 01 101 111	ED 6F	2	5	18	
RRD	[0347] [0347] A (HL)	?	?	?	0	?	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 flipflop.

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

Bit Manipulation 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
BIT b, r	$Z \leftarrow \overline{r_b}$	$?^1$	?	$?^2$	1	$?^3$	$?^4$	0	•	11 001 011 01 b r	CB	2	2	8	r Reg. 000 B
BIT b, (HL)	$Z \leftarrow \overline{(HL)_b}$	$?^1$	?	$?^2$	1	$?^3$	$?^4$	0	•	11 001 011 01 b 110	CB	2	3	12	001 C 010 D
BIT b, (IX + d)⁵	$Z \leftarrow \overline{(IX+d)_b}$	$?^1$	?	$?^2$	1	$?^3$	$?^4$	0	•	11 011 101 11 001 011 ? d ? 01 b 110	DD CB	4	5	20	011 E 100 H 101 L 111 A
BIT b, (IY + d)⁵	$Z \leftarrow \overline{(IY+d)_b}$	$?^1$	?	$?^2$	1	$?^3$	$?^4$	0	•	11 111 101 11 001 011 ? d ? 01 b 110	FD CB	4	5	20	
SET b, r	$r_b \leftarrow 1$	•	•	•	•	•	•	•	•	11 001 011 11 b r	CB	2	2	8	b Bit. 000 0 001 1
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	•	•	•	•	•	•	11 001 011 11 b 110	CB	2	4	15	010 2 011 3
SET b, (IX + d)	$(IX+d)_b \leftarrow 1$	•	•	•	•	•	•	•	•	11 011 101 11 001 011 ? d ? 11 b 110	DD CB	4	6	23	100 4 101 5 110 6 111 7
SET b, (IY + d)	$(IY+d)_b \leftarrow 1$	•	•	•	•	•	•	•	•	11 111 101 11 001 011 ? d ? 11 b 110	FD CB	4	6	23	
LD r,SET b, (IX + d)*	$r \leftarrow (IX+d)$ $r_b \leftarrow 1$ $(IX+d) \leftarrow r$	•	•	•	•	•	•	•	•	11 011 101 11 001 011 ? d ? 11 b r	DD CB	4	6	23	
LD r,SET b, (IY + d)*	$r \leftarrow (IY+d)$ $r_b \leftarrow 1$ $(IY+d) \leftarrow r$	•	•	•	•	•	•	•	•	11 111 101 11 001 011 ? d ? 11 b r	FD CB	4	6	23	
RES b, m	$m_b \leftarrow 0$ m ← r, (HL), (IX+d), (IY+d)	•	•	•	•	•	•	•	•	10					To form new opcode replace 11 of SET b, s with 10. Flags and states are the same.

Notes:
The notation m_b indicates bit b (0 to 7) of location m.
BIT instructions are performed by an bitwise AND.
[1] S is set if b = 7 and Z = 0
[2] F5 is set if b = 5 and Z = 0
[3] F3 is set if b = 3 and Z = 0
[4] P/V is set like the Z flag
[5] This instruction has other unofficial opcodes
* means unofficial instruction.

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

Input and Output Groups

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
IN A, (n)	A ← (n)	•	•	•	•	•	•	•	•	11 011 011 ? n ?	DB	2	3	11	r Reg. 000 B
IN r, (C)	r ← (C)	?	?	?	0	?	P	0	•	11 101 101 01 r 000	ED	2	3	12	001 C 010 D
IN (C)* or IN F, (C)*	Just affects flags, value is lost.	?	?	?	0	?	P	0	•	11 101 101 01 110 000	ED 70	2	3	12	011 E 100 H
INI	(HL) ← (C) HL ← HL + 1 B ← B - 1	?[1]	?[1]	?[1]	?[3]	?[1]	X	?[2]	?[3]	11 101 101 10 100 010	ED A2	2	4	16	101 L 111 A
INIR	(HL) ← (C) HL ← HL + 1 B ← B - 1 Repeat until B = 0	0	1	0	?[3]	0	X	?[2]	?[3]	11 101 101 10 110 010	ED B2	2 2	5 4	21 16	if B ≠ 0 if B = 0
IND	(HL) ← (C) HL ← HL - 1 B ← B - 1	?[1]	?[1]	?[1]	?[4]	?[1]	X	?[2]	?[4]	11 101 101 10 101 010	ED AA	2	4	16	
INDR	(HL) ← (C) HL ← HL - 1 B ← B - 1 Repeat until B = 0	0	1	0	?[4]	0	X	?[2]	?[4]	11 101 101 10 111 010	ED BA	2 2	5 4	21 16	if B ≠ 0 if B = 0
OUT (n), A	(n) ← A	•	•	•	•	•	•	•	•	11 010 011 ? n ?	D3	2	3	11	
OUT (C), r	(C) ← r	•	•	•	•	•	•	•	•	11 101 101 01 r 001	ED	2	3	12	
OUT (C), 0*	(C) ← 0	•	•	•	•	•	•	•	•	11 101 101 01 110 001	ED 71	2	3	12	
OUTI	(C) ← (HL) HL ← HL + 1 B ← B - 1	?[1]	?[1]	?[1]	X	?[1]	X	X	X	11 101 101 10 100 011	ED A3	2	4	16	
OTIR	(C) ← (HL) HL ← HL + 1 B ← B - 1 Repeat until B = 0	0	1	0	X	0	X	X	X	11 101 101 10 110 011	ED B3	2 2	5 4	21 16	if B ≠ 0 if B = 0
OUTD	(C) ← (HL) HL ← HL - 1 B ← B - 1	?[1]	?[1]	?[1]	X	?[1]	X	X	X	11 101 101 10 101 011	ED AB	2	4	16	
OTDR	(C) ← (HL) HL ← HL - 1 B ← B - 1 Repeat until B = 0	0	1	0	X	0	X	X	X	11 101 101 10 111 011	ED BB	2 2	5 4	21 16	if B ≠ 0 if B = 0

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.

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

[1] flag is affected by the result of B ← B - 1 as in DEC B.

[2] N is a copy bit 7 of the last value from the input (C).

[3] this flag contains the carry of (((C + 1) AND 255) + (C))

[4] this flag contains the carry of (((C - 1) AND 255) + (C))

* means unofficial instruction.

Flag Notation: • = flag is not affected, 0 = flag is reset, 1 = flag is set, X = flag is unknown,

? = flag is set according to the result of the operation.

Jump 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
JP nn	PC ? nn	•	•	•	•	•	•	•	•	11 000 011 ? n ? ? n ?	C3	3	3	10	
JP cc, nn	if cc is true, PC ? nn	•	•	•	•	•	•	•	•	11 ccc 010 ? n ? ? n ?		3	3	10	ccc Condition 000 NZ 001 Z 010 NC 011 C 100 PO 101 PE 110 P
JR e	PC ? PC + e	•	•	•	•	•	•	•	•	00 011 000 ? e - 2 ?	18	2	3	12	111 M
JR ss, e	if ss is true PC ? PC + e	•	•	•	•	•	•	•	•	00 ss 000 ? e - 2 ?		2 2	3 2	12 7	if ss is true if ss is false
JP HL	PC ? HL	•	•	•	•	•	•	•	•	11 101 001	E9	1	1	4	ss Condition
JP IX	PC ? IX	•	•	•	•	•	•	•	•	11 011 101 11 101 001	DD E9	2	2	8	111 C 110 NC
JP IY	PC ? IY	•	•	•	•	•	•	•	•	11 111 101 11 101 001	FD E9	2	2	8	101 Z 100 NZ
DJNZ e	B ? B - 1 if B ? 0 PC ? PC + e	•	•	•	•	•	•	•	•	00 010 000 ? e - 2 ?	10	2 2	2 3	8 13	if B = 0 if B ? 0

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, ? = flag is set according to the result of the operation.

Call and Return 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
CALL nn	SP ? SP - 1 (SP) ? PC_H SP ? SP - 1 (SP) ? PC_L PC ? nn	•	•	•	•	•	•	•	•	11 001 101 ? n ? ? n ?	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 ? n ? ? n ?		3 3	3 5	10 17	if cc is false if cc is true
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	1 3	5 11	if cc is false if cc is true
RETI[2]	PC_L ? (SP) SP ? SP + 1 PC_H ? (SP) SP ? SP + 1	•	•	•	•	•	•	•	•	11 101 101 01 001 101	ED 4D	2	4	14	cc Condition 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 01 000 101	ED 45	2	4	14	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	t 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 IFF_1 ? IFF_2

Flag Notation: • = flag is not affected, 0 = flag is reset, 1 = flag is set, ? = 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	45	LD B, L	8A	ADC A, D		
01 n n	LD BC, nn	46	LD B, (HL)	8B	ADC A, E		
02	LD (BC), A	47	LD B, A	8C	ADC A, H		
03	INC BC	48	LD C, B	8D	ADC A, L		
04	INC B	49	LD C, C	8E	ADC A, (HL)		
05	DEC B	4A	LD C, D	8F	ADC A, A		
06 n	LD B, n	4B	LD C, E	90	SUB B		
07	RLCA	4C	LD C, H	91	SUB C		
08	EX AF, AF'	4D	LD C, L	92	SUB D		
09	ADD HL, BC	4E	LD C, (HL)	93	SUB E		
0A	LD A, (BC)	4F	LD C, A	94	SUB H		
0B	DEC BC	50	LD D, B	95	SUB L		
0C	INC C	51	LD D, C	96	SUB (HL)		
0D	DEC C	52	LD D, D	97	SUB A		
0E n	LD C, n	53	LD D, E	98	SBC A, B		
0F	RRCA	54	LD D, H	99	SBC A, C		
10 n	DJNZ PC + n	55	LD D, L	9A	SBC A, D		
11 n n	LD DE, nn	56	LD D, (HL)	9B	SBC A, E		
12	LD (DE), A	57	LD D, A	9C	SBC A, H		
13	INC DE	58	LD E, B	9D	SBC A, L		
14	INC D	59	LD E, C	9E	SBC A, (HL)		
15	DEC D	5A	LD E, D	9F	SBC A, A		
16 n	LD D, n	5B	LD E, E	A0	AND B		
17	RLA	5C	LD E, H	A1	AND C		
18 n	JR PC + n	5D	LD E, L	A2	AND D		
19	ADD HL, DE	5E	LD E, (HL)	A3	AND E		
1A	LD A, (DE)	5F	LD E, A	A4	AND H		
1B	DEC DE	60	LD H, B	A5	AND L		
1C	INC E	61	LD H, C	A6	AND (HL)		
1D	DEC E	62	LD H, D	A7	AND A		
1E n	LD E, n	63	LD H, E	A8	XOR B		
1F	RRA	64	LD H, H	A9	XOR C		
20 n	JR NZ, PC + n	65	LD H, L	AA	XOR D		
21 n n	LD HL, nn	66	LD H, (HL)	AB	XOR E		
22 n n	LD (nn), HL	67	LD H, A	AC	XOR H		
23	INC HL	68	LD L, B	AD	XOR L		
24	INC H	69	LD L, C	AE	XOR (HL)		
25	DEC H	6A	LD L, D	AF	XOR A		
26 n	LD H, n	6B	LD L, E	B0	OR B		
27	DAA	6C	LD L, H	B1	OR C		
28 n	JR Z, PC + n	6D	LD L, L	B2	OR D		
29	ADD HL, HL	6E	LD L, (HL)	B3	OR E		
2A n n	LD HL, (nn)	6F	LD L, A	B4	OR H		
2B	DEC HL	70	LD (HL), B	B5	OR L		
2C	INC L	71	LD (HL), C	B6	OR (HL)		
2D	DEC L	72	LD (HL), D	B7	OR A		
2E n	LD L, n	73	LD (HL), E	B8	CP B		
2F	CPL	74	LD (HL), H	B9	CP C		
30 n	JR NC, PC + n	75	LD (HL), L	BA	CP D		
31 n n	LD SP, nn	76	HALT	BB	CP E		
32 n n	LD (nn), A	77	LD (HL), A	BC	CP H		
33	INC SP	78	LD A, B	BD	CP L		
34	INC (HL)	79	LD A, C	BE	CP (HL)		
35	DEC (HL)	7A	LD A, D	BF	CP A		
36 n	LD (HL), n	7B	LD A, E	C0	RET NZ		
37	SCF	7C	LD A, H	C1	POP BC		
38 n	JR C, PC + n	7D	LD A, L	C2 n n	JP NZ, nn		
39	ADD HL, SP	7E	LD A, (HL)	C3 n n	JP nn		
3A n n	LD A, (nn)	7F	LD A, A	C4 n n	CALL NZ, nn		
3B	DEC SP	80	ADD A, B	C5	PUSH BC		
3C	INC A	81	ADD A, C	C6 n	ADD A, n		
3D	DEC A	82	ADD A, D	C7	RST 0h		
3E n	LD A, n	83	ADD A, E	C8	RET Z		
3F	CCF	84	ADD A, H	C9	RET		
40	LD B, B	85	ADD A, L	CA n n	JP Z, nn		
41	LD B, C	86	ADD A, (HL)	CB00	RLC B		
42	LD B, D	87	ADD A, A	CB01	RLC C		
43	LD B, E	88	ADC A, B	CB02	RLC D		
44	LD B, H	89	ADC A, C	CB03	RLC E		

Code	Mnemonic	Code	Mnemonic	Code	Mnemonic
CB04	RLC H	CB56	BIT 2, (HL)	CBA8	RES 5, B
CB05	RLC L	CB57	BIT 2, A	CBA9	RES 5, C
CB06	RLC (HL)	CB58	BIT 3, B	CBAA	RES 5, D
CB07	RLC A	CB59	BIT 3, C	CBAB	RES 5, E
CB08	RRC B	CB5A	BIT 3, D	CBAC	RES 5, H
CB09	RRC C	CB5B	BIT 3, E	CBAD	RES 5, L
CB0A	RRC D	CB5C	BIT 3, H	CBAE	RES 5, (HL)
CB0B	RRC E	CB5D	BIT 3, L	CBAF	RES 5, A
CB0C	RRC H	CB5E	BIT 3, (HL)	CBB0	RES 6, B
CB0D	RRC L	CB5F	BIT 3, A	CBB1	RES 6, C
CB0E	RRC (HL)	CB60	BIT 4, B	CBB2	RES 6, D
CB0F	RRC A	CB61	BIT 4, C	CBB3	RES 6, E
CB10	RL B	CB62	BIT 4, D	CBB4	RES 6, H
CB11	RL C	CB63	BIT 4, E	CBB5	RES 6, L
CB12	RL D	CB64	BIT 4, H	CBB6	RES 6, (HL)
CB13	RL E	CB65	BIT 4, L	CBB7	RES 6, A
CB14	RL H	CB66	BIT 4, (HL)	CBB8	RES 7, B
CB15	RL L	CB67	BIT 4, A	CBB9	RES 7, C
CB16	RL (HL)	CB68	BIT 5, B	CBBA	RES 7, D
CB17	RL A	CB69	BIT 5, C	CBBB	RES 7, E
CB18	RR B	CB6A	BIT 5, D	CBBC	RES 7, H
CB19	RR C	CB6B	BIT 5, E	CBBD	RES 7, L
CB1A	RR D	CB6C	BIT 5, H	CBBE	RES 7, (HL)
CB1B	RR E	CB6D	BIT 5, L	CBBF	RES 7, A
CB1C	RR H	CB6E	BIT 5, (HL)	CBC0	SET 0, B
CB1D	RR L	CB6F	BIT 5, A	CBC1	SET 0, C
CB1E	RR (HL)	CB70	BIT 6, B	CBC2	SET 0, D
CB1F	RR A	CB71	BIT 6, C	CBC3	SET 0, E
CB20	SLA B	CB72	BIT 6, D	CBC4	SET 0, H
CB21	SLA C	CB73	BIT 6, E	CBC5	SET 0, L
CB22	SLA D	CB74	BIT 6, H	CBC6	SET 0, (HL)
CB23	SLA E	CB75	BIT 6, L	CBC7	SET 0, A
CB24	SLA H	CB76	BIT 6, (HL)	CBC8	SET 1, B
CB25	SLA L	CB77	BIT 6, A	CBC9	SET 1, C
CB26	SLA (HL)	CB78	BIT 7, B	CBCA	SET 1, D
CB27	SLA A	CB79	BIT 7, C	CBCB	SET 1, E
CB28	SRA B	CB7A	BIT 7, D	CBCC	SET 1, H
CB29	SRA C	CB7B	BIT 7, E	CBCD	SET 1, L
CB2A	SRA D	CB7C	BIT 7, H	CBCE	SET 1, (HL)
CB2B	SRA E	CB7D	BIT 7, L	CBCF	SET 1, A
CB2C	SRA H	CB7E	BIT 7, (HL)	CBD0	SET 2, B
CB2D	SRA L	CB7F	BIT 7, A	CBD1	SET 2, C
CB2E	SRA (HL)	CB80	RES 0, B	CBD2	SET 2, D
CB2F	SRA A	CB81	RES 0, C	CBD3	SET 2, E
CB30	SLL B*	CB82	RES 0, D	CBD4	SET 2, H
CB31	SLL C*	CB83	RES 0, E	CBD5	SET 2, L
CB32	SLL D*	CB84	RES 0, H	CBD6	SET 2, (HL)
CB33	SLL E*	CB85	RES 0, L	CBD7	SET 2, A
CB34	SLL H*	CB86	RES 0, (HL)	CBD8	SET 3, B
CB35	SLL L*	CB87	RES 0, A	CBD9	SET 3, C
CB36	SLL (HL)*	CB88	RES 1, B	CBDA	SET 3, D
CB37	SLL A*	CB89	RES 1, C	CBDB	SET 3, E
CB38	SRL B	CB8A	RES 1, D	CBDC	SET 3, H
CB39	SRL C	CB8B	RES 1, E	CBDD	SET 3, L
CB3A	SRL D	CB8C	RES 1, H	CBDE	SET 3, (HL)
CB3B	SRL E	CB8D	RES 1, L	CBDF	SET 3, A
CB3C	SRL H	CB8E	RES 1, (HL)	CBE0	SET 4, B
CB3D	SRL L	CB8F	RES 1, A	CBE1	SET 4, C
CB3E	SRL (HL)	CB90	RES 2, B	CBE2	SET 4, D
CB3F	SRL A	CB91	RES 2, C	CBE3	SET 4, E
CB40	BIT 0, B	CB92	RES 2, D	CBE4	SET 4, H
CB41	BIT 0, C	CB93	RES 2, E	CBE5	SET 4, L
CB42	BIT 0, D	CB94	RES 2, H	CBE6	SET 4, (HL)
CB43	BIT 0, E	CB95	RES 2, L	CBE7	SET 4, A
CB44	BIT 0, H	CB96	RES 2, (HL)	CBE8	SET 5, B
CB45	BIT 0, L	CB97	RES 2, A	CBE9	SET 5, C
CB46	BIT 0, (HL)	CB98	RES 3, B	CBEA	SET 5, D
CB47	BIT 0, A	CB99	RES 3, C	CBEB	SET 5, E
CB48	BIT 1, B	CB9A	RES 3, D	CBEC	SET 5, H
CB49	BIT 1, C	CB9B	RES 3, E	CBED	SET 5, L
CB4A	BIT 1, D	CB9C	RES 3, H	CBEE	SET 5, (HL)
CB4B	BIT 1, E	CB9D	RES 3, L	CBEF	SET 5, A
CB4C	BIT 1, H	CB9E	RES 3, (HL)	CBF0	SET 6, B
CB4D	BIT 1, L	CB9F	RES 3, A	CBF1	SET 6, C
CB4E	BIT 1, (HL)	CBA0	RES 4, B	CBF2	SET 6, D
CB4F	BIT 1, A	CBA1	RES 4, C	CBF3	SET 6, E
CB50	BIT 2, B	CBA2	RES 4, D	CBF4	SET 6, H
CB51	BIT 2, C	CBA3	RES 4, E	CBF5	SET 6, L
CB52	BIT 2, D	CBA4	RES 4, H	CBF6	SET 6, (HL)
CB53	BIT 2, E	CBA5	RES 4, L	CBF7	SET 6, A
CB54	BIT 2, H	CBA6	RES 4, (HL)	CBF8	SET 7, B
CB55	BIT 2, L	CBA7	RES 4, A	CBF9	SET 7, C

CBFA	SET 7, D	DD8C	ADC A, IX$_H$*
CBFB	SET 7, E	DD8D	ADC A, IX$_L$*
CBFC	SET 7, H	DD8E d	ADC A, (IX + d)
CBFD	SET 7, L	DD94	SUB IX$_H$*
CBFE	SET 7, (HL)	DD95	SUB IX$_L$*
CBFF	SET 7, A	DD96 d	SUB (IX + d)
CC n n	CALL Z, nn	DD9C	SBC A, IX$_H$*
CD n n	CALL nn	DD9D	SBC A, IX$_L$*
CE n	ADC A, n	DD9E d	SBC A, (IX + d)
CF	RST 8h	DDA4	AND IX$_H$*
D0	RET NC	DDA5	AND IX$_L$*
D1	POP DE	DDA6 d	AND (IX + d)
D2 n n	JP NC, nn	DDAC	XOR IX$_H$*
D3 n	OUT (n), A	DDAD	XOR IX$_L$*
D4 n n	CALL NC, nn	DDAE d	XOR (IX + d)
D5	PUSH DE	DDB4	OR IX$_H$*
D6 n	SUB n	DDB5	OR IX$_L$*
D7	RST 10h	DDB6 d	OR (IX + d)
D8	RETC	DDBC	CP IX$_H$*
D9	EXX	DDBD	CP IX$_L$*
DA n n	JP C, nn	DDBE d	CP (IX + d)
DB n	IN A, (n)	DDCB d 00	LD B, RLC (IX + d)*
DC n n	CALL C, nn	DDCB d 01	LD C, RLC (IX + d)*
DD09	ADD IX, BC	DDCB d 02	LD D, RLC (IX + d)*
DD19	ADD IX, DE	DDCB d 03	LD E, RLC (IX + d)*
DD21 n n	LD IX, nn	DDCB d 04	LD H, RLC (IX + d)*
DD22 n n	LD (nn), IX	DDCB d 05	LD L, RLC (IX + d)*
DD23	INC IX	DDCB d 06	RLC (IX + d)
DD24	INC IX$_H$*	DDCB d 07	LD A, RLC (IX + d)*
DD25	DEC IX$_H$*	DDCB d 08	LD B, RRC (IX + d)*
DD26 n	LD IX$_H$, n*	DDCB d 09	LD C, RRC (IX + d)*
DD29	ADD IX, IX	DDCB d 0A	LD D, RRC (IX + d)*
DD2A n n	LD IX, (nn)	DDCB d 0B	LD E, RRC (IX + d)*
DD2B	DEC IX	DDCB d 0C	LD H, RRC (IX + d)*
DD2C	INC IX$_L$*	DDCB d 0D	LD L, RRC (IX + d)*
DD2D	DEC IX$_L$*	DDCB d 0E	RRC (IX + d)
DD2E n	LD IX$_L$, n*	DDCB d 0F	LD A, RRC (IX + d)*
DD34 d	INC (IX + d)	DDCB d 10	LD B, RL (IX + d)*
DD35 d	DEC (IX + d)	DDCB d 11	LD C, RL (IX + d)*
DD36 d n	LD (IX + d), n	DDCB d 12	LD D, RL (IX + d)*
DD39	ADD IX, SP	DDCB d 13	LD E, RL (IX + d)*
DD44	LD B, IX$_H$*	DDCB d 14	LD H, RL (IX + d)*
DD45	LD B, IX$_L$*	DDCB d 15	LD L, RL (IX + d)*
DD46 d	LD B, (IX + d)	DDCB d 16	RL (IX + d)
DD4C	LD C, IX$_H$*	DDCB d 17	LD A, RL (IX + d)*
DD4D	LD C, IX$_L$*	DDCB d 18	LD B, RR (IX + d)*
DD4E d	LD C, (IX + d)	DDCB d 19	LD C, RR (IX + d)*
DD54	LD D, IX$_H$*	DDCB d 1A	LD D, RR (IX + d)*
DD55	LD D, IX$_L$*	DDCB d 1B	LD E, RR (IX + d)*
DD56 d	LD D, (IX + d)	DDCB d 1C	LD H, RR (IX + d)*
DD5C	LD E, IX$_H$*	DDCB d 1D	LD L, RR (IX + d)*
DD5D	LD E, IX$_L$*	DDCB d 1E	RR (IX + d)
DD5E d	LD E, (IX + d)	DDCB d 1F	LD A, RR (IX + d)*
DD60	LD IX$_H$, B*	DDCB d 20	LD B, SLA (IX + d)*
DD61	LD IX$_H$, C*	DDCB d 21	LD C, SLA (IX + d)*
DD62	LD IX$_H$, D*	DDCB d 22	LD D, SLA (IX + d)*
DD63	LD IX$_H$, E*	DDCB d 23	LD E, SLA (IX + d)*
DD64	LD IX$_H$, IX$_H$*	DDCB d 24	LD H, SLA (IX + d)*
DD65	LD IX$_H$, IX$_L$*	DDCB d 25	LD L, SLA (IX + d)*
DD66 d	LD H, (IX + d)	DDCB d 26	SLA (IX + d)
DD67	LD IX$_H$, A*	DDCB d 27	LD A, SLA (IX + d)*
DD68	LD IX$_L$, B*	DDCB d 28	LD B, SRA (IX + d)*
DD69	LD IX$_L$, C*	DDCB d 29	LD C, SRA (IX + d)*
DD6A	LD IX$_L$, D*	DDCB d 2A	LD D, SRA (IX + d)*
DD6B	LD IX$_L$, E*	DDCB d 2B	LD E, SRA (IX + d)*
DD6C	LD IX$_L$, IX$_H$*	DDCB d 2C	LD H, SRA (IX + d)*
DD6D	LD IX$_L$, IX$_L$*	DDCB d 2D	LD L, SRA (IX + d)*
DD6E d	LD L, (IX + d)	DDCB d 2E	SRA (IX + d)
DD6F	LD IX$_L$, A*	DDCB d 2F	LD A, SRA (IX + d)*
DD70 d	LD (IX + d), B	DDCB d 30	LD B, SLL (IX + d)*
DD71 d	LD (IX + d), C	DDCB d 31	LD C, SLL (IX + d)*
DD72 d	LD (IX + d), D	DDCB d 32	LD D, SLL (IX + d)*
DD73 d	LD (IX + d), E	DDCB d 33	LD E, SLL (IX + d)*
DD74 d	LD (IX + d), H	DDCB d 34	LD H, SLL (IX + d)*
DD75 d	LD (IX + d), L	DDCB d 35	LD L, SLL (IX + d)*
DD77 d	LD (IX + d), A	DDCB d 36	SLL (IX + d)*
DD7C	LD A, IX$_H$*	DDCB d 37	LD A, SLL (IX + d)*
DD7D	LD A, IX$_L$*	DDCB d 38	LD B, SRL (IX + d)*
DD7E d	LD A, (IX + d)	DDCB d 39	LD C, SRL (IX + d)*
DD84	ADD A, IX$_H$*	DDCB d 3A	LD D, SRL (IX + d)*
DD85	ADD A, IX$_L$*	DDCB d 3B	LD E, SRL (IX + d)*
DD86 d	ADD A, (IX + d)	DDCB d 3C	LD H, SRL (IX + d)*

DDCB d 3D	LD L, SRL (IX + d)*	DDCB d 65	BIT 4, (IX + d)*
DDCB d 3E	SRL (IX + d)	DDCB d 66	BIT 4, (IX + d)
DDCB d 3F	LD A, SRL (IX + d)*	DDCB d 67	BIT 4, (IX + d)*
DDCB d 40	BIT 0, (IX + d)*	DDCB d 68	BIT 5, (IX + d)*
DDCB d 41	BIT 0, (IX + d)*	DDCB d 69	BIT 5, (IX + d)*
DDCB d 42	BIT 0, (IX + d)*	DDCB d 6A	BIT 5, (IX + d)*
DDCB d 43	BIT 0, (IX + d)*	DDCB d 6B	BIT 5, (IX + d)*
DDCB d 44	BIT 0, (IX + d)*	DDCB d 6C	BIT 5, (IX + d)*
DDCB d 45	BIT 0, (IX + d)*	DDCB d 6D	BIT 5, (IX + d)*
DDCB d 46	BIT 0, (IX + d)	DDCB d 6E	BIT 5, (IX + d)
DDCB d 47	BIT 0, (IX + d)*	DDCB d 6F	BIT 5, (IX +d)*
DDCB d 48	BIT 1, (IX + d)*	DDCB d 70	BIT 6, (IX + d)*
DDCB d 49	BIT 1, (IX + d)*	DDCB d 71	BIT 6, (IX + d)*
DDCB d 4A	BIT 1, (IX + d)*	DDCB d 72	BIT 6, (IX + d)*
DDCB d 4B	BIT 1, (IX + d)*	DDCB d 73	BIT 6, (IX + d)*
DDCB d 4C	BIT 1, (IX + d)*	DDCB d 74	BIT 6, (IX + d)*
DDCB d 4D	BIT 1, (IX + d)*	DDCB d 75	BIT 6, (IX + d)*
DDCB d 4E	BIT 1, (IX + d)	DDCB d 76	BIT 6, (IX + d)
DDCB d 4F	BIT 1, (IX + d)*	DDCB d 77	BIT 6, (IX + d)*
DDCB d 50	BIT 2, (IX + d)*	DDCB d 78	BIT 7, (IX + d)*
DDCB d 51	BIT 2, (IX + d)*	DDCB d 79	BIT 7, (IX + d)*
DDCB d 52	BIT 2, (IX + d)*	DDCB d 7A	BIT 7, (IX + d)*
DDCB d 53	BIT 2, (IX + d)*	DDCB d 7B	BIT 7, (IX + d)*
DDCB d 54	BIT 2, (IX + d)*	DDCB d 7C	BIT 7, (IX + d)*
DDCB d 55	BIT 2, (IX + d)*	DDCB d 7D	BIT 7, (IX + d)*
DDCB d 56	BIT 2, (IX + d)	DDCB d 7E	BIT 7, (IX + d)
DDCB d 57	BIT 2, (IX + d)*	DDCB d 7F	BIT 7, (IX + d)*
DDCB d 58	BIT 3, (IX + d)*	DDCB d 80	LD B, RES 0, (IX + d)*
DDCB d 59	BIT 3, (IX + d)*	DDCB d 81	LD C, RES 0, (IX + d)*
DDCB d 5A	BIT 3, (IX + d)*	DDCB d 82	LD D, RES 0, (IX + d)*
DDCB d 5B	BIT 3, (IX + d)*	DDCB d 83	LD E, RES 0, (IX + d)*
DDCB d 5C	BIT 3, (IX + d)*	DDCB d 84	LD H, RES 0, (IX + d)*
DDCB d 5D	BIT 3, (IX + d)*	DDCB d 85	LD L, RES 0, (IX + d)*
DDCB d 5E	BIT 3, (IX + d)	DDCB d 86	RES 0, (IX + d)
DDCB d 5F	BIT 3, (IX + d)*	DDCB d 87	LD A, RES 0, (IX + d)*
DDCB d 60	BIT 4, (IX + d)*	DDCB d 88	LD B, RES 1, (IX + d)*
DDCB d 61	BIT 4, (IX + d)*	DDCB d 89	LD C, RES 1, (IX + d)*
DDCB d 62	BIT 4, (IX + d)*	DDCB d 8A	LD D, RES 1, (IX + d)*
DDCB d 63	BIT 4, (IX + d)*	DDCB d 8B	LD E, RES 1, (IX + d)*
DDCB d 64	BIT 4, (IX + d)*	DDCB d 8C	LD H, RES 1, (IX + d)*
		DDCB d 8D	LD L, RES 1, (IX + d)*
		DDCB d 8E	RES 1, (IX + d)

Opcode	Mnemonic	Opcode	Mnemonic	Opcode	Mnemonic
DDCB d 8F	LD A, RES 1, (IX + d)*	DDCB d E1	LD C, SET 4, (IX + d)*	ED5F	LD A, R
DDCB d 90	LD B, RES 2, (IX + d)*	DDCB d E2	LD D, SET 4, (IX + d)*	ED60	IN H, (C)
DDCB d 91	LD C, RES 2, (IX + d)*	DDCB d E3	LD E, SET 4, (IX + d)*	ED61	OUT (C), H
DDCB d 92	LD D, RES 2, (IX + d)*	DDCB d E4	LD H, SET 4, (IX + d)*	ED62	SBC HL, HL
DDCB d 93	LD E, RES 2, (IX + d)*	DDCB d E5	LD L, SET 4, (IX + d)*	ED63 n n	LD (nn), HL
DDCB d 94	LD H, RES 2, (IX + d)*	DDCB d E6	SET 4, (IX + d)	ED64	NEG*
DDCB d 95	LD L, RES 2, (IX + d)*	DDCB d E7	LD A, SET 4, (IX + d)*	ED65	RETN*
DDCB d 96	RES 2, (IX + d)	DDCB d E8	LD B, SET 5, (IX + d)*	ED66	IM 0*
DDCB d 97	LD A, RES 2, (IX + d)*	DDCB d E9	LD C, SET 5, (IX + d)*	ED67	RRD
DDCB d 98	LD B, RES 3, (IX + d)*	DDCB d EA	LD D, SET 5, (IX + d)*	ED68	IN L, (C)
DDCB d 99	LD C, RES 3, (IX + d)*	DDCB d EB	LD E, SET 5, (IX + d)*	ED69	OUT (C), L
DDCB d 9A	LD D, RES 3, (IX + d)*	DDCB d EC	LD H, SET 5, (IX + d)*	ED6A	ADC HL, HL
DDCB d 9B	LD E, RES 3, (IX + d)*	DDCB d ED	LD L, SET 5, (IX + d)*	ED6B n n	LD HL, (nn)
DDCB d 9C	LD H, RES 3, (IX + d)*	DDCB d EE	SET 5, (IX + d)	ED6C	NEG*
DDCB d 9D	LD L, RES 3, (IX + d)*	DDCB d EF	LD A, SET 5, (IX + d)*	ED6D	RETN*
DDCB d 9E	RES 3, (IX + d)	DDCB d F0	LD B, SET 6, (IX + d)*	ED6E	IM 0/1*
DDCB d 9F	LD A, RES 3, (IX + d)*	DDCB d F1	LD C, SET 6, (IX + d)*	ED6F	RLD
DDCB d A0	LD B, RES 4, (IX + d)*	DDCB d F2	LD D, SET 6, (IX + d)*	ED70	IN (C)* / IN F, (C)*
DDCB d A1	LD C, RES 4, (IX + d)*	DDCB d F3	LD E, SET 6, (IX + d)*	ED71	OUT (C), 0*
DDCB d A2	LD D, RES 4, (IX + d)*	DDCB d F4	LD H, SET 6, (IX + d)*	ED72	SBC HL, SP
DDCB d A3	LD E, RES 4, (IX + d)*	DDCB d F5	LD L, SET 6, (IX + d)*	ED73 n n	LD (nn), SP
DDCB d A4	LD H, RES 4, (IX + d)*	DDCB d F6	SET 6, (IX + d)	ED74	NEG*
DDCB d A5	LD L, RES 4, (IX + d)*	DDCB d F7	LD A, SET 6, (IX + d)*	ED75	RETN*
DDCB d A6	RES 4, (IX + d)	DDCB d F8	LD B, SET 7, (IX + d)*	ED76	IM 1*
DDCB d A7	LD A, RES 4, (IX + d)*	DDCB d F9	LD C, SET 7, (IX + d)*	ED78	IN A, (C)
DDCB d A8	LD B, RES 5, (IX + d)*	DDCB d FA	LD D, SET 7, (IX + d)*	ED79	OUT (C), A
DDCB d A9	LD C, RES 5, (IX + d)*	DDCB d FB	LD E, SET 7, (IX + d)*	ED7A	ADC HL, SP
DDCB d AA	LD D, RES 5, (IX + d)*	DDCB d FC	LD H, SET 7, (IX + d)*	ED7B n n	LD SP, (nn)
DDCB d AB	LD E, RES 5, (IX + d)*	DDCB d FD	LD L, SET 7, (IX + d)*	ED7C	NEG*
DDCB d AC	LD H, RES 5, (IX + d)*	DDCB d FE	SET 7, (IX + d)	ED7D	RETN*
DDCB d AD	LD L, RES 5, (IX + d)*	DDCB d FF	LD A, SET 7, (IX + d)*	ED7E	IM 2*
DDCB d AE	RES 5, (IX + d)	DDE1	POP IX	EDA0	LDI
DDCB d AF	LD A, RES 5, (IX + d)*	DDE3	EX (SP), IX	EDA1	CPI
DDCB d B0	LD B, RES 6, (IX + d)*	DDE5	PUSH IX	EDA2	INI
DDCB d B1	LD C, RES 6, (IX + d)*	DDE9	JP (IX)	EDA3	OUTI
DDCB d B2	LD D, RES 6, (IX + d)*	DDF9	LD SP, IX	EDA8	LDD
DDCB d B3	LD E, RES 6, (IX + d)*	DE n	SBC A, n	EDA9	CPD
DDCB d B4	LD H, RES 6, (IX + d)*	DF	RST 18h	EDAA	IND
DDCB d B5	LD L, RES 6, (IX + d)*	E0	RET PO	EDAB	OUTD
DDCB d B6	RES 6, (IX + d)	E1	POP HL	EDB0	LDIR
DDCB d B7	LD A, RES 6, (IX + d)*	E2 n n	JP PO, nn	EDB1	CPIR
DDCB d B8	LD B, RES 7, (IX + d)*	E3	EX (SP), HL	EDB2	INIR
DDCB d B9	LD C, RES 7, (IX + d)*	E4 n n	CALL PO, nn	EDB3	OTIR
DDCB d BA	LD D, RES 7, (IX + d)*	E5	PUSH HL	EDB8	LDDR
DDCB d BB	LD E, RES 7, (IX + d)*	E6 n	AND n	EDB9	CPDR
DDCB d BC	LD H, RES 7, (IX + d)*	E7	RST 20h	EDBA	INDR
DDCB d BD	LD L, RES 7, (IX + d)*	E8	RET PE	EDBB	OTDR
DDCB d BE	RES 7, (IX + d)	E9	JP (HL)	EE n	XOR n
DDCB d BF	LD A, RES 7, (IX + d)*	EA n n	JP PE, (nn)	EF	RST 28h
DDCB d C0	LD B, SET 0, (IX + d)*	EB	EX DE, HL	F0	RET P
DDCB d C1	LD C, SET 0, (IX + d)*	EC n n	CALL PE, nn	F1	POP AF
DDCB d C2	LD D, SET 0, (IX + d)*	ED40	IN B, (C)	F2 n n	JP P, nn
DDCB d C3	LD E, SET 0, (IX + d)*	ED41	OUT (C), B	F3	DI
DDCB d C4	LD H, SET 0, (IX + d)*	ED42	SBC HL, BC	F4 n n	CALL P, nn
DDCB d C5	LD L, SET 0, (IX + d)*	ED43 n n	LD (nn), BC	F5	PUSH AF
DDCB d C6	SET 0, (IX + d)	ED44	NEG	F6 n	OR n
DDCB d C7	LD A, SET 0, (IX + d)*	ED45	RETN	F7	RST 30h
DDCB d C8	LD B, SET 1, (IX + d)*	ED46	IM 0	F8	RET M
DDCB d C9	LD C, SET 1, (IX + d)*	ED47	LD I, A	F9	LD SP, HL
DDCB d CA	LD D, SET 1, (IX + d)*	ED48	IN C, (C)	FA n n	JP M, nn
DDCB d CB	LD E, SET 1, (IX + d)*	ED49	OUT (C), C	FB	EI
DDCB d CC	LD H, SET 1, (IX + d)*	ED4A	ADC HL, BC	FC n n	CALL M, nn
DDCB d CD	LD L, SET 1, (IX + d)*	ED4B n n	LD BC, (nn)	FD09	ADD IY, BC
DDCB d CE	SET 1, (IX + d)	ED4C	NEG*	FD19	ADD IY, DE
DDCB d CF	LD A, SET 1, (IX + d)*	ED4D	RETI	FD21 n n	LD IY, nn
DDCB d D0	LD B, SET 2, (IX + d)*	ED4E	IM 0/1*	FD22 n n	LD (nn), IY
DDCB d D1	LD C, SET 2, (IX + d)*	ED4F	LD R, A	FD23	INC IY
DDCB d D2	LD D, SET 2, (IX + d)*	ED50	IN D, (C)	FD24	INC IY_H*
DDCB d D3	LD E, SET 2, (IX + d)*	ED51	OUT (C), D	FD25	DEC IY_H*
DDCB d D4	LD H, SET 2, (IX + d)*	ED52	SBC HL, DE	FD26 n	LD IY_H, n*
DDCB d D5	LD L, SET 2, (IX + d)*	ED53 n n	LD (nn), DE	FD29	ADD IY, IY
DDCB d D6	SET 2, (IX + d)	ED54	NEG*	FD2A n n	LD IY, (nn)
DDCB d D7	LD A, SET 2, (IX + d)*	ED55	RETN*	FD2B	DEC IY
DDCB d D8	LD B, SET 3, (IX + d)*	ED56	IM 1	FD2C	INC IY_L*
DDCB d D9	LD C, SET 3, (IX + d)*	ED57	LD A, I	FD2D	DEC IY_L*
DDCB d DA	LD D, SET 3, (IX + d)*	ED58	IN E, (C)	FD2E n	LD IY_L, n*
DDCB d DB	LD E, SET 3, (IX + d)*	ED59	OUT (C), E	FD34 d	INC (IY + d)
DDCB d DC	LD H, SET 3, (IX + d)*	ED5A	ADC HL, DE	FD35 d	DEC (IY + d)
DDCB d DD	LD L, SET 3, (IX + d)*	ED5B n n	LD DE, (nn)	FD36 d n	LD (IY + d), n
DDCB d DE	SET 3, (IX + d)	ED5C	NEG*	FD39	ADD IY, SP
DDCB d DF	LD A, SET 3, (IX + d)*	ED5D	RETN*	FD44	LD B, IY_H*
DDCB d E0	LD B, SET 4, (IX + d)*	ED5E	IM 2	FD45	LD B, IY_L*

Opcode	Instruction
FD46 d	LD B, (IY + d)
FD4C	LD C, IY_H*
FD4D	LD C, IY_L*
FD4E d	LD C, (IY + d)
FD54	LD D, IY_H*
FD55	LD D, IY_L*
FD56 d	LD D, (IY + d)
FD5C	LD E, IY_H*
FD5D	LD E, IY_L*
FD5E d	LD E, (IY + d)
FD60	LD IY_H, B*
FD61	LD IY_H, C*
FD62	LD IY_H, D*
FD63	LD IY_H, E*
FD64	LD IY_H, IY_H*
FD65	LD IY_H, IY_L*
FD66 d	LD H, (IY + d)
FD67	LD IY_H, A*
FD68	LD IY_L, B*
FD69	LD IY_L, C*
FD6A	LD IY_L, D*
FD6B	LD IY_L, E*
FD6C	LD IY_L, IY_H*
FD6D	LD IY_L, IY_L*
FD6E d	LD L, (IY + d)
FD6F	LD IY_L, A*
FD70 d	LD (IY + d), B
FD71 d	LD (IY + d), C
FD72 d	LD (IY + d), D
FD73 d	LD (IY + d), E
FD74 d	LD (IY + d), H
FD75 d	LD (IY + d), L
FD77 d	LD (IY + d), A
FD7C	LD A, IY_H*
FD7D	LD A, IY_L*
FD7E d	LD A, (IY + d)
FD84	ADD A, IY_H*
FD85	ADD A, IY_L*
FD86 d	ADD A, (IY + d)
FD8C	ADC A, IY_H*
FD8D	ADC A, IY_L*
FD8E d	ADC A, (IY + d)
FD94	SUB IY_H*
FD95	SUB IY_L*
FD96 d	SUB (IY + d)
FD9C	SBC A, IY_H*
FD9D	SBC A, IY_L*
FD9E d	SBC A, (IY + d)
FDA4	AND IY_H*
FDA5	AND IY_L*
FDA6 d	AND (IY + d)
FDAC	XOR IY_H*
FDAD	XOR IY_L*
FDAE d	XOR (IY + d)
FDB4	OR IY_H*
FDB5	OR IY_L*
FDB6 d	OR (IY + d)
FDBC	CP IY_H*
FDBD	CP IY_L*
FDBE d	CP (IY + d)
FDCB d 00	LD B, RLC (IY + d)*
FDCB d 01	LD C, RLC (IY + d)*
FDCB d 02	LD D, RLC (IY + d)*
FDCB d 03	LD E, RLC (IY + d)*
FDCB d 04	LD H, RLC (IY + d)*
FDCB d 05	LD L, RLC (IY + d)*
FDCB d 06	RLC (IY + d)
FDCB d 07	LD A, RLC (IY + d)*
FDCB d 08	LD B, RRC (IY + d)*
FDCB d 09	LD C, RRC (IY + d)*
FDCB d 0A	LD D, RRC (IY + d)*
FDCB d 0B	LD E, RRC (IY + d)*
FDCB d 0C	LD H, RRC (IY + d)*
FDCB d 0D	LD L, RRC (IY + d)*
FDCB d 0E	RRC (IY + d)
FDCB d 0F	LD A, RRC (IY + d)*
FDCB d 10	LD B, RL (IY + d)*
FDCB d 11	LD C, RL (IY + d)*
FDCB d 12	LD D, RL (IY + d)*
FDCB d 13	LD E, RL (IY + d)*
FDCB d 14	LD H, RL (IY + d)*
FDCB d 15	LD L, RL (IY + d)*
FDCB d 16	RL (IY + d)
FDCB d 17	LD A, RL (IY + d)*
FDCB d 18	LD B, RR (IY + d)*
FDCB d 19	LD C, RR (IY + d)*
FDCB d 1A	LD D, RR (IY + d)*
FDCB d 1B	LD E, RR (IY + d)*
FDCB d 1C	LD H, RR (IY + d)*
FDCB d 1D	LD L, RR (IY + d)*
FDCB d 1E	RR (IY + d)
FDCB d 1F	LD A, RR (IY + d)*
FDCB d 20	LD B, SLA (IY + d)*
FDCB d 21	LD C, SLA (IY + d)*
FDCB d 22	LD D, SLA (IY + d)*
FDCB d 23	LD E, SLA (IY + d)*
FDCB d 24	LD H, SLA (IY + d)*
FDCB d 25	LD L, SLA (IY + d)*
FDCB d 26	SLA (IY + d)
FDCB d 27	LD A, SLA (IY + d)*
FDCB d 28	LD B, SRA (IY + d)*
FDCB d 29	LD C, SRA (IY + d)*
FDCB d 2A	LD D, SRA (IY + d)*
FDCB d 2B	LD E, SRA (IY + d)*
FDCB d 2C	LD H, SRA (IY + d)*
FDCB d 2D	LD L, SRA (IY + d)*
FDCB d 2E	SRA (IY + d)
FDCB d 2F	LD A, SRA (IY + d)*
FDCB d 30	LD B, SLL (IY + d)*
FDCB d 31	LD C, SLL (IY + d)*
FDCB d 32	LD D, SLL (IY + d)*
FDCB d 33	LD E, SLL (IY + d)*
FDCB d 34	LD H, SLL (IY + d)*
FDCB d 35	LD L, SLL (IY + d)*
FDCB d 36	SLL (IY + d)*
FDCB d 37	LD A, SLL (IY + d)*
FDCB d 38	LD B, SRL (IY + d)*
FDCB d 39	LD C, SRL (IY + d)*
FDCB d 3A	LD D, SRL (IY + d)*
FDCB d 3B	LD E, SRL (IY + d)*
FDCB d 3C	LD H, SRL (IY + d)*
FDCB d 3D	LD L, SRL (IY + d)*
FDCB d 3E	SRL (IY + d)
FDCB d 3F	LD A, SRL (IY + d)*
FDCB d 40	BIT 0, (IY + d)*
FDCB d 41	BIT 0, (IY + d)*
FDCB d 42	BIT 0, (IY + d)*
FDCB d 43	BIT 0, (IY + d)*
FDCB d 44	BIT 0, (IY + d)*
FDCB d 45	BIT 0, (IY + d)*
FDCB d 46	BIT 0, (IY + d)
FDCB d 47	BIT 0, (IY + d)*
FDCB d 48	BIT 1, (IY + d)*
FDCB d 49	BIT 1, (IY + d)*
FDCB d 4A	BIT 1, (IY + d)*
FDCB d 4B	BIT 1, (IY + d)*
FDCB d 4C	BIT 1, (IY + d)*
FDCB d 4D	BIT 1, (IY + d)*
FDCB d 4E	BIT 1, (IY + d)
FDCB d 4F	BIT 1, (IY + d)*
FDCB d 50	BIT 2, (IY + d)*
FDCB d 51	BIT 2, (IY + d)*
FDCB d 52	BIT 2, (IY + d)*
FDCB d 53	BIT 2, (IY + d)*
FDCB d 54	BIT 2, (IY + d)*
FDCB d 55	BIT 2, (IY + d)*
FDCB d 56	BIT 2, (IY + d)
FDCB d 57	BIT 2, (IY + d)*
FDCB d 58	BIT 3, (IY + d)*
FDCB d 59	BIT 3, (IY + d)*
FDCB d 5A	BIT 3, (IY + d)*
FDCB d 5B	BIT 3, (IY + d)*
FDCB d 5C	BIT 3, (IY + d)*
FDCB d 5D	BIT 3, (IY + d)*
FDCB d 5E	BIT 3, (IY + d)
FDCB d 5F	BIT 3, (IY + d)*
FDCB d 60	BIT 4, (IY + d)*
FDCB d 61	BIT 4, (IY + d)*
FDCB d 62	BIT 4, (IY + d)*
FDCB d 63	BIT 4, (IY + d)*
FDCB d 64	BIT 4, (IY + d)*
FDCB d 65	BIT 4, (IY + d)*
FDCB d 66	BIT 4, (IY + d)
FDCB d 67	BIT 4, (IY + d)*
FDCB d 68	BIT 5, (IY + d)*
FDCB d 69	BIT 5, (IY + d)*
FDCB d 6A	BIT 5, (IY + d)*
FDCB d 6B	BIT 5, (IY + d)*
FDCB d 6C	BIT 5, (IY + d)*
FDCB d 6D	BIT 5, (IY + d)*
FDCB d 6E	BIT 5, (IY + d)
FDCB d 6F	BIT 5, (IY + d)*
FDCB d 70	BIT 6, (IY + d)*
FDCB d 71	BIT 6, (IY + d)*
FDCB d 72	BIT 6, (IY + d)*
FDCB d 73	BIT 6, (IY + d)*
FDCB d 74	BIT 6, (IY + d)*
FDCB d 75	BIT 6, (IY + d)*
FDCB d 76	BIT 6, (IY + d)
FDCB d 77	BIT 6, (IY + d)*
FDCB d 78	BIT 7, (IY + d)*
FDCB d 79	BIT 7, (IY + d)*
FDCB d 7A	BIT 7, (IY + d)*
FDCB d 7B	BIT 7, (IY + d)*
FDCB d 7C	BIT 7, (IY + d)*
FDCB d 7D	BIT 7, (IY + d)*
FDCB d 7E	BIT 7, (IY + d)
FDCB d 7F	BIT 7, (IY + d)*
FDCB d 80	LD B, RES 0, (IY + d)*
FDCB d 81	LD C, RES 0, (IY + d)*
FDCB d 82	LD D, RES 0, (IY + d)*
FDCB d 83	LD E, RES 0, (IY + d)*
FDCB d 84	LD H, RES 0, (IY + d)*
FDCB d 85	LD L, RES 0, (IY + d)*
FDCB d 86	RES 0, (IY + d)
FDCB d 87	LD A, RES 0, (IY + d)*
FDCB d 88	LD B, RES 1, (IY + d)*
FDCB d 89	LD C, RES 1, (IY + d)*
FDCB d 8A	LD D, RES 1, (IY + d)*
FDCB d 8B	LD E, RES 1, (IY + d)*
FDCB d 8C	LD H, RES 1, (IY + d)*
FDCB d 8D	LD L, RES 1, (IY + d)*
FDCB d 8E	RES 1, (IY + d)
FDCB d 8F	LD A, RES 1, (IY + d)*
FDCB d 90	LD B, RES 2, (IY + d)*
FDCB d 91	LD C, RES 2, (IY + d)*
FDCB d 92	LD D, RES 2, (IY + d)*
FDCB d 93	LD E, RES 2, (IY + d)*
FDCB d 94	LD H, RES 2, (IY + d)*
FDCB d 95	LD L, RES 2, (IY + d)*
FDCB d 96	RES 2, (IY + d)
FDCB d 97	LD A, RES 2, (IY + d)*
FDCB d 98	LD B, RES 3, (IY + d)*
FDCB d 99	LD C, RES 3, (IY + d)*
FDCB d 9A	LD D, RES 3, (IY + d)*
FDCB d 9B	LD E, RES 3, (IY + d)*
FDCB d 9C	LD H, RES 3, (IY + d)*
FDCB d 9D	LD L, RES 3, (IY + d)*
FDCB d 9E	RES 3, (IY + d)
FDCB d 9F	LD A, RES 3, (IY + d)*
FDCB d A0	LD B, RES 4, (IY + d)*
FDCB d A1	LD C, RES 4, (IY + d)*
FDCB d A2	LD D, RES 4, (IY + d)*
FDCB d A3	LD E, RES 4, (IY + d)*
FDCB d A4	LD H, RES 4, (IY + d)*
FDCB d A5	LD L, RES 4, (IY + d)*
FDCB d A6	RES 4, (IY + d)
FDCB d A7	LD A, RES 4, (IY + d)*
FDCB d A8	LD B, RES 5, (IY + d)*
FDCB d A9	LD C, RES 5, (IY + d)*
FDCB d AA	LD D, RES 5, (IY + d)*
FDCB d AB	LD E, RES 5, (IY + d)*
FDCB d AC	LD H, RES 5, (IY + d)*
FDCB d AD	LD L, RES 5, (IY + d)*
FDCB d AE	RES 5, (IY + d)
FDCB d AF	LD A, RES 5, (IY + d)*
FDCB d B0	LD B, RES 6, (IY + d)*
FDCB d B1	LD C, RES 6, (IY + d)*
FDCB d B2	LD D, RES 6, (IY + d)*
FDCB d B3	LD E, RES 6, (IY + d)*
FDCB d B4	LD H, RES 6, (IY + d)*
FDCB d B5	LD L, RES 6, (IY + d)*
FDCB d B6	RES 6, (IY + d)
FDCB d B7	LD A, RES 6, (IY + d)*
FDCB d B8	LD B, RES 7, (IY + d)*
FDCB d B9	LD C, RES 7, (IY + d)*

```
FDCB d BA    LD D, RES 7, (IY + d)*
FDCB d BB    LD E, RES 7, (IY + d)*
FDCB d BC    LD H, RES 7, (IY + d)*
FDCB d BD    LD L, RES 7, (IY + d)*
FDCB d BE    RES 7, (IY + d)
FDCB d BF    LD A, RES 7, (IY + d)*
FDCB d C0    LD B, SET 0, (IY + d)*
FDCB d C1    LD C, SET 0, (IY + d)*
FDCB d C2    LD D, SET 0, (IY + d)*
FDCB d C3    LD E, SET 0, (IY + d)*
FDCB d C4    LD H, SET 0, (IY + d)*
FDCB d C5    LD L, SET 0, (IY + d)*
FDCB d C6    SET 0, (IY + d)
FDCB d C7    LD A, SET 0, (IY + d)*
FDCB d C8    LD B, SET 1, (IY + 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
```