

Z80 And R800 Opcodes List

Opcodes on single-byte

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NOP	LD BC,ad	LD (BC),A	INC BC	INC B	DEC B	LD B,v	RLCA	EX AF,AF'	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DEC C	LD C,v	RRCA
10	DJNZ v	LD DE,ad	LD (DE),A	INC DE	INC D	DEC D	LD D,v	RLA	JR v	ADD HL,DE	LD A,(DE)	DEC DE	INC E	DEC E	LD E,v	RRA
20	JR nz,v	LD HL,ad	LD (ad),HL	INC HL	INC H	DEC H	LD H,v	DAA	JR z,v	ADD HL,HL	LD HL,(ad)	DEC HL	INC L	DEC L	LD L,v	CPL
30	JR nc,v	LD SP,ad	LD (ad),A	INC SP	INC (HL)	DEC (HL)	LD (HL),v	SCF	JR C,v	ADD HL,SP	LD A,(ad)	DEC SP	INC A	DEC A	LD A,v	CCF
40	LD B,B	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	LD C,C	LD C,D	LD C,E	LD C,H	LD C,L	LD C,(HL)	LD C,A
50	LD D,B	LD D,C	LD D,D	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	LD E,E	LD E,H	LD E,L	LD E,(HL)	LD E,A
60	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD L,L	LD L,(HL)	LD L,A
70	LD (HL),B	LD (HL),C	LD (HL),D	LD (HL),E	LD (HL),H	LD (HL),L	HALT	LD (HL),A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,(HL)	LD A,A
80	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)	ADD A,A	ADC A,B	ADC A,C	ADC A,D	ADC A,E	ADC A,H	ADC A,L	ADC A,(HL)	ADC A,A
90	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB (HL)	SUB A	SBC A,B	SBC A,C	SBC A,D	SBC A,E	SBC A,H	SBC A,L	SBC A,(HL)	SBC A
A0	AND B	AND C	AND D	AND E	AND H	AND L	AND (HL)	AND A	XOR B	XOR C	XOR D	XOR E	XOR H	XOR L	XOR (HL)	XOR A
B0	OR B	OR C	OR D	OR E	OR H	OR L	OR (HL)	OR A	CP B	CP C	CP D	CP E	CP H	CP L	CP (HL)	CP A
C0	RET nz	POP BC	JP nz,ad	JP ad	CALL nz,ad	PUSH BC	ADD A,v	RST 0h	RET z	RET	JP z,ad	*	CALL z,ad	CALL ad	ADC A,v	RST 8h
D0	RET nc	POP DE	JP nc,ad	OUT (v),A	CALL nc,ad	PUSH DE	SUB v	RST 10h	RET c	EXX	JP c,ad	IN A,(v)	CALL c,ad	**	SBC A,v	RST 18h
E0	RET po	POP HL	JP po,ad	EX (SP),HL	CALL po,ad	PUSH HL	AND v	RST 20h	RET pe	JP (HL)	JP pe,ad	EX DE,HL	CALL pe,ad	***	XOR v	RST 28h
F0	RET p	POP AF	JP p,ad	DI	CALL p,ad	PUSH AF	OR v	RST 30h	RET m	LD SP,HL	JP m,ad	EI	CALL m,ad	****	CP v	RST 38h

Note: Some of these opcodes are followed by one or two bytes to specify the value v or the address ad.

(*) Opcodes extended on 2-byte which begin with CBh

CB	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	RLC B	RLC C	RLC D	RLC E	RLC H	RLC L	RLC (HL)	RLC A	RRC B	RRC C	RRC D	RRC E	RRC H	RRC H	RRC (HL)	RRC A
10	RL B	RL C	RL D	RL E	RL H	RL L	RL (HL)	RL A	RR B	RR C	RR D	RR E	RR H	RR H	RR (HL)	RR A
20	SLA B	SLA C	SLA D	SLA E	SLA H	SLA L	SLA (HL)	SLA A	SRA B	SRA C	SRA D	SRA E	SRA H	SRA H	SRA (HL)	SRA A
30	SLL B	SLL C	SLL D	SLL E	SLL H	SLL L	SLL (HL)	SLL A	SRL B	SRL C	SRL D	SRL E	SRL H	SRL H	SRL (HL)	SRL A
40	BIT 0,B	BIT 0,C	BIT 0,D	BIT 0,E	BIT 0,H	BIT 0,L	BIT 0,(HL)	BIT 0,A	BIT 1,B	BIT 1,C	BIT 1,D	BIT 1,E	BIT 1,H	BIT 1,L	BIT 1,(HL)	BIT 1,A
50	BIT 2,B	BIT 2,C	BIT 2,D	BIT 2,E	BIT 2,H	BIT 2,L	BIT 2,(HL)	BIT 2,A	BIT 3,B	BIT 3,C	BIT 3,D	BIT 3,E	BIT 3,H	BIT 3,L	BIT 3,(HL)	BIT 3,A
60	BIT 4,B	BIT 4,C	BIT 4,D	BIT 4,E	BIT 4,H	BIT 4,L	BIT 4,(HL)	BIT 4,A	BIT 5,B	BIT 5,C	BIT 5,D	BIT 5,E	BIT 5,H	BIT 5,L	BIT 5,(HL)	BIT 5,A
70	BIT 6,B	BIT 6,C	BIT 6,D	BIT 6,E	BIT 6,H	BIT 6,L	BIT 6,(HL)	BIT 6,A	BIT 7,B	BIT 7,C	BIT 7,D	BIT 7,E	BIT 7,H	BIT 7,L	BIT 7,(HL)	BIT 7,A
80	RES 0,B	RES 0,C	RES 0,D	RES 0,E	RES 0,H	RES 0,L	RES 0,(HL)	RES 0,A	RES 1,B	RES 1,C	RES 1,D	RES 1,E	RES 1,H	RES 1,L	RES 1,(HL)	RES 1,A
90	RES 2,B	RES 2,C	RES 2,D	RES 2,E	RES 2,H	RES 2,L	RES 2,(HL)	RES 2,A	RES 3,B	RES 3,C	RES 3,D	RES 3,E	RES 3,H	RES 3,L	RES 3,(HL)	RES 3,A
A0	RES 4,B	RES 4,C	RES 4,D	RES 4,E	RES 4,H	RES 4,L	RES 4,(HL)	RES 4,A	RES 5,B	RES 5,C	RES 5,D	RES 5,E	RES 5,H	RES 5,L	RES 5,(HL)	RES 5,A
B0	RES 6,B	RES 6,C	RES 6,D	RES 6,E	RES 6,H	RES 6,L	RES 6,(HL)	RES 6,A	RES 7,B	RES 7,C	RES 7,D	RES 7,E	RES 7,H	RES 7,L	RES 7,(HL)	RES 7,A
C0	SET 0,B	SET 0,C	SET 0,D	SET 0,E	SET 0,H	SET 0,L	SET 0,(HL)	SET 0,A	SET 1,B	SET 1,C	SET 1,D	SET 1,E	SET 1,H	SET 1,L	SET 1,(HL)	SET 1,A
D0	SET 2,B	SET 2,C	SET 2,D	SET 2,E	SET 2,H	SET 2,L	SET 2,(HL)	SET 2,A	SET 3,B	SET 3,C	SET 3,D	SET 3,E	SET 3,H	SET 3,L	SET 3,(HL)	SET 3,A
E0	SET 4,B	SET 4,C	SET 4,D	SET 4,E	SET 4,H	SET 4,L	SET 4,(HL)	SET 4,A	SET 5,B	SET 5,C	SET 5,D	SET 5,E	SET 5,H	SET 5,L	SET 5,(HL)	SET 5,A
F0	SET 6,B	SET 6,C	SET 6,D	SET 6,E	SET 6,H	SET 6,L	SET 6,(HL)	SET 6,A	SET 7,B	SET 7,C	SET 7,D	SET 7,E	SET 7,H	SET 7,L	SET 7,(HL)	SET 7,A

(**) Opcodes extended on two-byte which begin with DDh (for register IX)

DD	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00										ADD IX,BC						
10										ADD IX,DE						
20		LD IX,ad	LD (ad),IX	INC IX	INC IXH	DEC IXH	LD IXH,v			ADD IX,IX	LD IX,(ad)	DEC IX	INC IXL	DEC IXL	LD IXL,v	
30					INC (IX+v)	DEC (IX+v)	LD (IX+v),v			ADD IX,SP						
40					LD B,IXH	LD B,IXL	LD B,(IX+v)						LD C,IXH	LD C,IXL	LD C,(IX+v)	
50					LD D,IXH	LD D,IXL	LD D,(IX+v)						LD E,IXH	LD E,IXL	LD E,(IX+v)	
60	LD IXH,B	LD IXH,C	LD IXH,D	LD IXH,E	LD IXH,IXH	LD IXH,IXL	LD H,(IX+v)	LD IXH,A	LD IXL,B	LD IXL,C	LD IXL,D	LD IXL,E	LD IXL,IXH	LD IXL,IXL	LD L,(IX+v)	LD IXL,A
70	LD (IX+v),B	LD (IX+v),C	LD (IX+v),D	LD (IX+v),E	LD (IX+v),H	LD (IX+v),L		LD (IX+v),A					LD A,IXH	LD A,IXL	LD A,(IX+v)	
80					ADD A,IXH	ADD A,IXL	ADD A,(IX+v)						ADC A,IXH	ADC A,IXL	ADC A,(IX+v)	
90					SUB IXH	SUB IXL	SUB (IX+v)						SBC A,IXH	SBC A,IXL	SBC A,(IX+v)	
A0					AND IXH	AND IXL	AND (IX+v)						XOR IXH	XOR IXL	XOR (IX+v)	
B0					OR IXH	OR IXL	OR (IX+v)						CP IXH	CP IXL	CP (IX+v)	
C0												**2				
D0																
E0		POP IX		EX (SP),IX		PUSH IX				JP (IX)						
F0										LD SP,IX						

Note: Some of these opcodes are followed by one or two bytes to specify the value v or the address ad.

(**2) Opcodes extended on four-byte which begin with DDh+CBh+v (for register IX)

DD+CB+v	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	RLC (v+IX) LD B,(v+IX)	RLC (IX+v) LD C,(IX+v)	RLC (v+IX) LD D,(v+IX)	RLC (v+IX) LD E,(v+IX)	RLC (v+IX) LD H,(v+IX)	RLC (v+IX) LD L,(v+IX)	RLC (v+IX) LD B,(v+IX)	RLC (v+IX) LD A,(v+IX)	RRC (v+IX) LD B,(v+IX)	RRC (v+IX) LD C,(v+IX)	RRC (v+IX) LD D,(v+IX)	RRC (v+IX) LD E,(v+IX)	RRC (v+IX) LD H,(v+IX)	RRC (v+IX) LD L,(v+IX)	RRC (v+IX) LD A,(v+IX)	RRC (v+IX) LD A,(v+IX)
10	RL (v+IX) LD B,(v+IX)	RL (IX+v) LD C,(IX+v)	RL (v+IX) LD D,(v+IX)	RL (v+IX) LD E,(v+IX)	RL (v+IX) LD H,(v+IX)	RL (v+IX) LD L,(v+IX)	RL (v+IX) LD B,(v+IX)	RL (v+IX) LD A,(v+IX)	RR (v+IX) LD B,(v+IX)	RR (v+IX) LD C,(v+IX)	RR (v+IX) LD D,(v+IX)	RR (v+IX) LD E,(v+IX)	RR (v+IX) LD H,(v+IX)	RR (v+IX) LD L,(v+IX)	RR (v+IX) LD A,(v+IX)	RR (v+IX) LD A,(v+IX)
20	SLA (v+IX) LD B,(v+IX)	SLA (IX+v) LD C,(IX+v)	SLA (v+IX) LD D,(v+IX)	SLA (v+IX) LD E,(v+IX)	SLA (v+IX) LD H,(v+IX)	SLA (v+IX) LD L,(v+IX)	SLA (v+IX) LD B,(v+IX)	SLA (v+IX) LD A,(v+IX)	SRA (v+IX) LD B,(v+IX)	SRA (v+IX) LD C,(v+IX)	SRA (v+IX) LD D,(v+IX)	SRA (v+IX) LD E,(v+IX)	SRA (v+IX) LD H,(v+IX)	SRA (v+IX) LD L,(v+IX)	SRA (v+IX) LD A,(v+IX)	SRA (v+IX) LD A,(v+IX)
30	SLL (v+IX) LD B,(v+IX)	SLL (IX+v) LD C,(IX+v)	SLL (v+IX) LD D,(v+IX)	SLL (v+IX) LD E,(v+IX)	SLL (v+IX) LD H,(v+IX)	SLL (v+IX) LD L,(v+IX)	SLL (v+IX) LD B,(v+IX)	SLL (v+IX) LD A,(v+IX)	SRL (v+IX) LD B,(v+IX)	SRL (v+IX) LD C,(v+IX)	SRL (v+IX) LD D,(v+IX)	SRL (v+IX) LD E,(v+IX)	SRL (v+IX) LD H,(v+IX)	SRL (v+IX) LD L,(v+IX)	SRL (v+IX) LD A,(v+IX)	SRL (v+IX) LD A,(v+IX)
40	BIT 0,(v+IX) LD B,(v+IX)	BIT 0,(IX+v) LD C,(IX+v)	BIT 0,(v+IX) LD D,(v+IX)	BIT 0,(v+IX) LD E,(v+IX)	BIT 0,(v+IX) LD H,(v+IX)	BIT 0,(v+IX) LD L,(v+IX)	BIT 0,(v+IX) LD B,(v+IX)	BIT 0,(v+IX) LD A,(v+IX)	BIT 1,(v+IX) LD B,(v+IX)	BIT 1,(v+IX) LD C,(v+IX)	BIT 1,(v+IX) LD D,(v+IX)	BIT 1,(v+IX) LD E,(v+IX)	BIT 1,(v+IX) LD H,(v+IX)	BIT 1,(v+IX) LD L,(v+IX)	BIT 1,(v+IX) LD A,(v+IX)	BIT 1,(v+IX) LD A,(v+IX)
50	BIT 2,(v+IX) LD B,(v+IX)	BIT 2,(IX+v) LD C,(IX+v)	BIT 2,(v+IX) LD D,(v+IX)	BIT 2,(v+IX) LD E,(v+IX)	BIT 2,(v+IX) LD H,(v+IX)	BIT 2,(v+IX) LD L,(v+IX)	BIT 2,(v+IX) LD B,(v+IX)	BIT 2,(v+IX) LD A,(v+IX)	BIT 3,(v+IX) LD B,(v+IX)	BIT 3,(v+IX) LD C,(v+IX)	BIT 3,(v+IX) LD D,(v+IX)	BIT 3,(v+IX) LD E,(v+IX)	BIT 3,(v+IX) LD H,(v+IX)	BIT 3,(v+IX) LD L,(v+IX)	BIT 3,(v+IX) LD A,(v+IX)	BIT 3,(v+IX) LD A,(v+IX)
60	BIT 4,(v+IX) LD B,(v+IX)	BIT 4,(IX+v) LD C,(IX+v)	BIT 4,(v+IX) LD D,(v+IX)	BIT 4,(v+IX) LD E,(v+IX)	BIT 4,(v+IX) LD H,(v+IX)	BIT 4,(v+IX) LD L,(v+IX)	BIT 4,(v+IX) LD B,(v+IX)	BIT 4,(v+IX) LD A,(v+IX)	BIT 5,(v+IX) LD B,(v+IX)	BIT 5,(v+IX) LD C,(v+IX)	BIT 5,(v+IX) LD D,(v+IX)	BIT 5,(v+IX) LD E,(v+IX)	BIT 5,(v+IX) LD H,(v+IX)	BIT 5,(v+IX) LD L,(v+IX)	BIT 5,(v+IX) LD A,(v+IX)	BIT 5,(v+IX) LD A,(v+IX)
70	BIT 6,(v+IX) LD B,(v+IX)	BIT 6,(IX+v) LD C,(IX+v)	BIT 6,(v+IX) LD D,(v+IX)	BIT 6,(v+IX) LD E,(v+IX)	BIT 6,(v+IX) LD H,(v+IX)	BIT 6,(v+IX) LD L,(v+IX)	BIT 6,(v+IX) LD B,(v+IX)	BIT 6,(v+IX) LD A,(v+IX)	BIT 7,(v+IX) LD B,(v+IX)	BIT 7,(v+IX) LD C,(v+IX)	BIT 7,(v+IX) LD D,(v+IX)	BIT 7,(v+IX) LD E,(v+IX)	BIT 7,(v+IX) LD H,(v+IX)	BIT 7,(v+IX) LD L,(v+IX)	BIT 7,(v+IX) LD A,(v+IX)	BIT 7,(v+IX) LD A,(v+IX)
80	RES 0,(v+IX) LD B,(v+IX)	RES 0,(IX+v) LD C,(IX+v)	RES 0,(v+IX) LD D,(v+IX)	RES 0,(v+IX) LD E,(v+IX)	RES 0,(v+IX) LD H,(v+IX)	RES 0,(v+IX) LD L,(v+IX)	RES 0,(v+IX) LD B,(v+IX)	RES 0,(v+IX) LD A,(v+IX)	RES 1,(v+IX) LD B,(v+IX)	RES 1,(v+IX) LD C,(v+IX)	RES 1,(v+IX) LD D,(v+IX)	RES 1,(v+IX) LD E,(v+IX)	RES 1,(v+IX) LD H,(v+IX)	RES 1,(v+IX) LD L,(v+IX)	RES 1,(v+IX) LD A,(v+IX)	RES 1,(v+IX) LD A,(v+IX)
90	RES 2,(v+IX) LD B,(v+IX)	RES 2,(IX+v) LD C,(IX+v)	RES 2,(v+IX) LD D,(v+IX)	RES 2,(v+IX) LD E,(v+IX)	RES 2,(v+IX) LD H,(v+IX)	RES 2,(v+IX) LD L,(v+IX)	RES 2,(v+IX) LD B,(v+IX)	RES 2,(v+IX) LD A,(v+IX)	RES 3,(v+IX) LD B,(v+IX)	RES 3,(v+IX) LD C,(v+IX)	RES 3,(v+IX) LD D,(v+IX)	RES 3,(v+IX) LD E,(v+IX)	RES 3,(v+IX) LD H,(v+IX)	RES 3,(v+IX) LD L,(v+IX)	RES 3,(v+IX) LD A,(v+IX)	RES 3,(v+IX) LD A,(v+IX)
A0	RES 4,(v+IX) LD B,(v+IX)	RES 4,(IX+v) LD C,(IX+v)	RES 4,(v+IX) LD D,(v+IX)	RES 4,(v+IX) LD E,(v+IX)	RES 4,(v+IX) LD H,(v+IX)	RES 4,(v+IX) LD L,(v+IX)	RES 4,(v+IX) LD B,(v+IX)	RES 4,(v+IX) LD A,(v+IX)	RES 5,(v+IX) LD B,(v+IX)	RES 5,(v+IX) LD C,(v+IX)	RES 5,(v+IX) LD D,(v+IX)	RES 5,(v+IX) LD E,(v+IX)	RES 5,(v+IX) LD H,(v+IX)	RES 5,(v+IX) LD L,(v+IX)	RES 5,(v+IX) LD A,(v+IX)	RES 5,(v+IX) LD A,(v+IX)
B0	RES 6,(v+IX) LD B,(v+IX)	RES 6,(IX+v) LD C,(IX+v)	RES 6,(v+IX) LD D,(v+IX)	RES 6,(v+IX) LD E,(v+IX)	RES 6,(v+IX) LD H,(v+IX)	RES 6,(v+IX) LD L,(v+IX)	RES 6,(v+IX) LD B,(v+IX)	RES 6,(v+IX) LD A,(v+IX)	RES 7,(v+IX) LD B,(v+IX)	RES 7,(v+IX) LD C,(v+IX)	RES 7,(v+IX) LD D,(v+IX)	RES 7,(v+IX) LD E,(v+IX)	RES 7,(v+IX) LD H,(v+IX)	RES 7,(v+IX) LD L,(v+IX)	RES 7,(v+IX) LD A,(v+IX)	RES 7,(v+IX) LD A,(v+IX)
C0	SET 0,(v+IX) LD B,(v+IX)	SET 0,(IX+v) LD C,(IX+v)	SET 0,(v+IX) LD D,(v+IX)	SET 0,(v+IX) LD E,(v+IX)	SET 0,(v+IX) LD H,(v+IX)	SET 0,(v+IX) LD L,(v+IX)	SET 0,(v+IX) LD B,(v+IX)	SET 0,(v+IX) LD A,(v+IX)	SET 1,(v+IX) LD B,(v+IX)	SET 1,(v+IX) LD C,(v+IX)	SET 1,(v+IX) LD D,(v+IX)	SET 1,(v+IX) LD E,(v+IX)	SET 1,(v+IX) LD H,(v+IX)	SET 1,(v+IX) LD L,(v+IX)	SET 1,(v+IX) LD A,(v+IX)	SET 1,(v+IX) LD A,(v+IX)
D0	SET 2,(v+IX) LD B,(v+IX)	SET 2,(IX+v) LD C,(IX+v)	SET 2,(v+IX) LD D,(v+IX)	SET 2,(v+IX) LD E,(v+IX)	SET 2,(v+IX) LD H,(v+IX)	SET 2,(v+IX) LD L,(v+IX)	SET 2,(v+IX) LD B,(v+IX)	SET 2,(v+IX) LD A,(v+IX)	SET 3,(v+IX) LD B,(v+IX)	SET 3,(v+IX) LD C,(v+IX)	SET 3,(v+IX) LD D,(v+IX)	SET 3,(v+IX) LD E,(v+IX)	SET 3,(v+IX) LD H,(v+IX)	SET 3,(v+IX) LD L,(v+IX)	SET 3,(v+IX) LD A,(v+IX)	SET 3,(v+IX) LD A,(v+IX)
E0	SET 4,(v+IX) LD B,(v+IX)	SET 4,(IX+v) LD C,(IX+v)	SET 4,(v+IX) LD D,(v+IX)	SET 4,(v+IX) LD E,(v+IX)	SET 4,(v+IX) LD H,(v+IX)	SET 4,(v+IX) LD L,(v+IX)	SET 4,(v+IX) LD B,(v+IX)	SET 4,(v+IX) LD A,(v+IX)	SET 5,(v+IX) LD B,(v+IX)	SET 5,(v+IX) LD C,(v+IX)	SET 5,(v+IX) LD D,(v+IX)	SET 5,(v+IX) LD E,(v+IX)	SET 5,(v+IX) LD H,(v+IX)	SET 5,(v+IX) LD L,(v+IX)	SET 5,(v+IX) LD A,(v+IX)	SET 5,(v+IX) LD A,(v+IX)
F0	SET 6,(v+IX) LD B,(v+IX)	SET 6,(IX+v) LD C,(IX+v)	SET 6,(v+IX) LD D,(v+IX)	SET 6,(v+IX) LD E,(v+IX)	SET 6,(v+IX) LD H,(v+IX)	SET 6,(v+IX) LD L,(v+IX)	SET 6,(v+IX) LD B,(v+IX)	SET 6,(v+IX) LD A,(v+IX)	SET 7,(v+IX) LD B,(v+IX)	SET 7,(v+IX) LD C,(v+IX)	SET 7,(v+IX) LD D,(v+IX)	SET 7,(v+IX) LD E,(v+IX)	SET 7,(v+IX) LD H,(v+IX)	SET 7,(v+IX) LD L,(v+IX)	SET 7,(v+IX) LD A,(v+IX)	SET 7,(v+IX) LD A,(v+IX)

Note: These opcodes have a byte between the third and fourth bytes to specify the value (v).

(***) Opcodes extended on two-byte which begin with EDh

ED	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00																
10																
20																
30																
40	IN B,(C)	OUT (C),B	SBC HL,BC	LD (ad),BC	NEG	RETN	IM 0	LD i,A	IN C,(C)	OUT (C),C	ADC HL,BC	LD BC,(ad)	NEG	RETI	IM 0	LD R,A
50	IN D,(C)	OUT (C),D	SBC DE,BC	LD (ad),DE	NEG	RETN	IM 1	LD A,i	IN E,(C)	OUT (C),E	ADC HL,DE	LD DE,(ad)	NEG	RETI	IM 2	LD A,R
60	IN H,(C)	OUT (C),H	SBC HL,HL	LD (ad),HL	NEG	RETN	IM 0	RRD	IN L,(C)	OUT (C),L	ADC HL,HL	LD HL,(ad)	NEG	RETI	IM 0	RLD
70	IN (C)	OUT (C),0	SBC HL,SP	LD (ad),SP	NEG	RETN	IM 1	NOP	IN A,(C)	OUT (C),A	ADC HL,SP	LD SP,(ad)	NEG	RETI	IM 2	LD R,R
80																
90																
A0	LDI	CPI	INI	OUTI					LDD	CPD	IND	OUTD				
B0	LDIR	CPIR	INIR	OTIR					LDDR	CPDR	INDR	OUTR				
C0		MULUB A,B		MULUW HL,BC						MULUB A,C						
D0		MULUB A,D		MULUW HL,DE						MULUB A,E						
E0		MULUB A,H		MULUW HL,HL						MULUB A,L						
F0				MULUW HL,SP						MULUB A,A						

Note: Some of these opcodes are followed by two bytes to specify an address (ad).

(****) Opcodes extended on two or three byte which begin with FDh (register IY)

See the tables of extended opcodes which begin with DDh and DDh+CBh. Replace DDh by FDh to obtain the code of instructions for register IY.

Description of colors

- The **green** instructions refer to inputs/outputs.
- The **purple** instructions concern interruptions.
- The instructions in **red** are undocumented by Zilog for the Z80. Some Z80-compatible CPUs do not have all of these instructions.
- The instructions in **sky blue** are unsigned multiplications. These are specific instructions for the R800 (MSX turbo R). **MULUB** multiplies the two indicated 8-bit registers and sets the result in HL. **MULUW** multiplies both 16-bit registers and sets the result in HL (MSB) and DE (LSB). Barred instructions are not guarantees of operation.

About Undocumented Instructions (in red)

- Undocumented instructions are rarely used in MSX programs.
- The instruction **SLL** of opcodes CBh 30h ~ CBh 37h (in dark red) has been replaced on R800 by another undocumented instruction which is similar to the instruction TEST of the Z280.
- **IN (C)** is an official instruction on R800. This instruction reads the I/O port indicated by the register C without keeping the result but the register F will be modified.
- On Z80 the undocumented opcodes which begin with DDh CBh or FDh CBh correspond to the execution of the two instructions indicated in the cell at same time.
- Empty cells correspond to a NOP whose execution takes 8T states and increment by two the register R. Same for the mirrors of NEG, RETN and IM (in italic and darker red). Same also for opcodes which begin with DDh CBh and FDH CBh.
- All instructions for the registers IXL, IYH and IYL are official instructions on R800.