

Contents

ADC	2	JRE	44	POP	91	SETUSR	130
ADD	3	LCALL	45	PUMA	92	SETUSRP	131
AESIMC	5	LD	46	PUMS	93	SHAF1	132
AESISR	6	LDD	54	PUSH	94	SHAF2	133
AESMC	7	LDDR	55	RDMODE	95	SHAF3	134
AESSR	8	LDDSR	56	RES	96	SL1REG	135
ALTD	9	LDF	57	RET	97	SL2REG	136
ALTS	10	LDI	58	RETI	98	SL3REG	137
ALTSD	11	LDIR	59	RL	99	SL4REG	138
AND	12	LDISR	60	RL1REG	100	SL5REG	139
BIT	14	LDL	61	RL2REG	101	SL6REG	140
BOOL	15	LDP	62	RL3REG	102	SL7REG	141
CALL	16	LJP	63	RL4REG	103	SL8REG	142
CBM	17	LLCALL	64	RL5REG	104	SLA	143
CCF	18	LLJP	65	RL6REG	105	SLL	144
CLR	19	LLRET	66	RL7REG	106	SR1REG	145
CNVC	20	LRET	67	RL8REG	107	SR2REG	146
CNVD	21	LSDDR	68	RLA	108	SR3REG	147
CONVC	22	LSDR	69	RLC	109	SR4REG	148
CONVD	23	LSIDR	70	RLCA	110	SR5REG	149
COPY	24	LSIR	71	RR	111	SR6REG	150
COPYR	25	MD5F1	72	RR1REG	112	SR7REG	151
CP	26	MD5F2	73	RR2REG	113	SR8REG	152
CPL	28	MD5F3	74	RR3REG	114	SRA	153
DEC	29	MUL	75	RR4REG	115	SRET	154
DJNZ	30	MULU	76	RR5REG	116	SRL	155
DWJNZ	31	NEG	77	RR6REG	117	SUB	156
EX	32	NOP	78	RR7REG	118	SURES	158
EXP	33	OR	79	RR8REG	119	SWAP	159
EXX	34	PLDD	81	RRA	120	SYSCALL	160
FLAG	35	PLDDR	82	RRC	121	TEST	161
IBOX	36	PLDDSR	83	RRCA	122	TSTNULL	162
INC	37	PLDI	84	RST	123	UMA	163
IOE	38	PLDIR	85	SBC	124	UMS	164
IOI	39	PLDISR	86	SBOX	125	XOR	165
IP	40	PLSDDR	87	SCALL	126	ZDMA	167
IPRES	41	PLSDR	88	SCF	127	ZINTACK	168
JP	42	PLSIDR	89	SET	128		
JR	43	PLSIR	90	SETSYSP	129		

ADC

20 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
ADC A, (HL)	10001110 A = A + (HL) + CF	5 (2,1,2)	5	5	5	*	*	V	*	fr	fr	s
ADC A, (IX+d)	11011101 10001110 ----d--- A = A + (IX+d) + CF	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	fr	s
ADC A, (IY+d)	11111101 10001110 ----d--- A = A + (IY+d) + CF	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	fr	s
ADC A, n	11001110 ----n--- A = A + n + CF	4 (2,2)	4	2	4	*	*	V	*	fr	fr	
ADC A, r	10001-r- A = A + r + CF	2	2	2	2	*	*	V	*	fr	fr	
ADC HL, ss	11101101 01ss1010 HL = HL + ss + CF	4 (2,2)	4	2	4	*	*	V	*	fr	fr	
ADC A, (HL)	01111111 10001110 A = A + (HL) + CF	7 (2,2,1,2)	7	5	7	*	*	V	*	fr	fr	s
ADC A, r	01111111 10001-r- A = A + r + CF	4 (2,2)	4	2	4	*	*	V	*	fr	fr	
ADC JKHL, BCDE	11101101 11001110 JKHL = JKHL + BCDE + CY	4 (2,2)	4	2	4	*	*	V	*	fr	fr	
ADC HL, (IX+d)	11011101 10000001 ----d--- HL = HL + (IX+d) + CF	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	fr	s
ADC JKHL, (IX+d)	11011101 10000011 ----d--- JKHL = JKHL + (IX+d) + CF	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	fr	s
ADC HL, (IY+d)	11111101 10000001 ----d--- HL = HL + (IY+d) + CF	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	fr	s
ADC JKHL, (IY+d)	11111101 10000011 ----d--- JKHL = JKHL + (IY+d) + CF	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	fr	s
ADC JKHL, ps	01001001 000100ps JKHL = JKHL + ps + CF	4 (2,2)	4	2	4	*	*	V	*	fr	fr	
ADC A, (ps+d)	01001001 000101ps ----d--- A = A + (ps+d) + CF	10 (2,2,2,2,2,2)	10	9	8	*	*	V	*	fr	fr	
ADC HL, (ps+d)	01001001 000110ps ----d--- HL = HL + (ps+d) + CF	12 (2,2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	fr	
ADC JKHL, (ps+d)	01001001 000111ps ----d--- JKHL = JKHL + (ps+d) + CF	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	fr	
ADC A, (sp+n)	01001001 10011001 ----n--- A = A + (sp+n) + CF	10 (2,2,2,2,2,2)	10	9	8	*	*	V	*	fr	fr	
ADC HL, (sp+n)	01001001 10011010 ----n--- HL = HL + (sp+n) + CF	12 (2,2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	fr	
ADC JKHL, (sp+n)	01001001 10011011 ----n--- JKHL = JKHL + (sp+n) + CF	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	fr	

ADD

27 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
ADD A, (HL)	10000110 A = A + (HL)	5 (2,1,2)	5	5	5	*	*	V	*	fr	r	s
ADD A, (IX+d)	11011101 10000110 ----d---	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	r	s
ADD A, (IY+d)	11111101 10000110 ----d---	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	r	s
ADD A, n	11000110 ----n---	4 (2,2)	4	2	4	*	*	V	*	fr	r	
ADD A, r	10000-r- A = A + r	2	2	2	2	*	*	V	*	fr	r	
ADD HL, ss	00ss1001 HL = HL + ss	2	2	2	2	-	-	-	*	fr	r	
ADD IX, xx	11011101 00xx1001 IX = IX + xx	4 (2,2)	4	2	4	-	-	-	*	f	r	
ADD IY, yy	11111101 00yy1001 IY = IY + yy	4 (2,2)	4	2	4	-	-	-	*	f	r	
ADD SP, d	00100111 ----d---	4 (2,2)	4	2	4	-	-	-	*	f		
ADD HL, JK	01100101 HL = HL + JK	2	2	2	2	-	-	-	*	fr	r	
ADD A, (HL)	01111111 10000110 A = A + (HL)	7 (2,2,1,2)	7	5	7	*	*	V	*	fr	r	s
ADD A, r	01111111 10000-r- A = A + r	4 (2,2)	4	2	4	*	*	V	*	fr	r	
ADD JKHL, BCDE	11101101 11000110 JKHL = JKHL + BCDE	4 (2,2)	4	2	4	-	-	-	*	fr	r	
ADD HL, JK	01111111 01100101 HL = HL + JK	4 (2,2)	4	2	4	-	-	-	*	fr	r	
ADD HL, (IX+d)	11011101 10000000 ----d---	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	r	s
ADD JKHL, (IX+d)	11011101 10000010 ----d---	12 (2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	r	s
ADD IX, d	11011101 11000101 ----d---	6 (2,2,2)	6	4	4	-	-	-	*	f		
ADD HL, (IY+d)	11111101 10000000 ----d---	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	r	s
ADD JKHL, (IY+d)	11111101 10000010 ----d---	12 (2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	r	s
ADD IY, d	11111101 11000101 ----d---	6 (2,2,2)	6	4	4	-	-	-	*	f		
ADD JKHL, ps	01001001 000000ps JKHL = JKHL + ps	4 (2,2)	4	2	4	*	*	V	*	fr	r	

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
ADD A, (ps+d)	01001001 000001ps ----d--- A = A + (ps+d)	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	r	
ADD HL, (ps+d)	01001001 000010ps ----d--- HL = HL + (ps+d)	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	r	
ADD JKHL, (ps+d)	01001001 000011ps ----d--- JKHL = JKHL + (ps+d)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	r	
ADD A, (sp+n)	01001001 10001001 ----n--- A = A + (sp+n)	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	r	
ADD HL, (sp+n)	01001001 10001010 ----n--- HL = HL + (sp+n)	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	r	
ADD JKHL, (sp+n)	01001001 10001011 ----n--- JKHL = JKHL + (sp+n)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	r	

AESIMC

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
AESIMC	01001001 10110011 aes inverse mix columns (PW, PX, PY and PZ)	4 (2,2)	4	2	4	-	-	-	-			

AESISR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
AESISR	01001001 10100011 aes inverse shift rows (PW, PX, PY and PZ)	4 (2,2)	4	2	4	-	-	-	-			

AESMC

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
AESMC	01001001 10110010 aes mix columns (PW, PX, PY and PZ)	4 (2,2)	4	2	4	-	-	-	-			

AESSR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
AESSR	01001001 10100010 aes shift rows (PW, PX, PY and PZ)	4 (2,2)	4	2	4	-	-	-	-			

ALTD

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
ALTD	01110110 alternate register destination for next instruction	2	2	2	2	-	-	-	-			

ALTS

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
ALTS	01000000 alternate register source for next instruction	2	2	2	2	-	-	-	-			

ALTSD

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
ALTSD	01100100 alternate register source and destination for next instruction	2	2	2	2	-	-	-	-			

AND

22 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
AND (HL)	10100110 A = A & (HL)	5 (2,1,2)	5	5	5	*	*	P	0	fr	r	s
AND (IX+d)	11011101 10100110 ----d---	10 (2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	s
AND (IY+d)	11111101 10100110 ----d---	10 (2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	s
AND HL,DE	11011100 HL = HL & DE	2	2	2	2	*	*	P	0	fr	r	
AND IX,DE	11011101 11011100 IX = IX & DE	4 (2,2)	4	2	4	*	*	P	0	f	r	
AND IY,DE	11111101 11011100 IY = IY & DE	4 (2,2)	4	2	4	*	*	P	0	f	r	
AND n	11100110 ----n---	4 (2,2)	4	2	4	*	*	P	0	fr	r	
AND r	10100-r- A = A & r	2	2	2	2	*	*	P	0	fr	r	
AND (HL)	01111111 10100110 A = A & (HL)	7 (2,2,1,2)	7	5	7	*	*	P	0	fr	r	s
AND r	01111111 10100-r- A = A & r	4 (2,2)	4	2	4	*	*	P	0	fr	r	
AND JKHL,BCDE	11101101 11100110 JKHL = JKHL & BCDE	4 (2,2)	4	2	4	*	*	P	0	fr	r	
AND HL,(IX+d)	11011101 10100000 ----d---	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	s
AND JKHL,(IX+d)	11011101 10100010 ----d---	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr	r	s
AND HL,(IY+d)	11111101 10100000 ----d---	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	s
AND JKHL,(IY+d)	11111101 10100010 ----d---	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr	r	s
AND JKHL,ps	01001001 010000ps JKHL = JKHL & ps	4 (2,2)	4	2	4	*	*	P	0	fr	r	
AND A,(ps+d)	01001001 010001ps ----d---	10 (2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	
AND HL,(ps+d)	01001001 010010ps ----d---	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	
AND JKHL,(ps+d)	01001001 010011ps ----d---	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr	r	
AND A,(sp+n)	01001001 11001001 ----n---	10 (2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	
AND HL,(sp+n)	01001001 11001010 ----n---	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
AND JKHL, (sp+n)	01001001 11001011 ----n--- JKHL = JKHL & (sp+n)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr		r

BIT

4 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
BIT b, (HL)	11001011 01-b-110 (HL) & bit	8 (2,2,2,2)	7	5	7	-	*	-	-	f		s
BIT b, (IX+d)	11011101 11001011 ----d--- 01-b-110 (IX+d) & bit	12 (2,2,2,2,2,2)	12	8	10	-	*	-	-	f		s
BIT b, (IY+d)	11111101 11001011 ----d--- 01-b-110 (IY+d) & bit	12 (2,2,2,2,2,2)	12	8	10	-	*	-	-	f		s
BIT b, r	11001011 01-b--r- r & bit	4 (2,2)	4	2	4	-	*	-	-	f	r	

B00L

3 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
B00L HL	11001100 if (HL != 0) HL = 1	2	2	2	2	*	*	0	0	fr	r	
B00L IX	11011101 11001100 if (IX != 0) IX = 1	4 (2,2)	4	2	4	*	*	0	0	f		
B00L IY	11111101 11001100 if (IY != 0) IY = 1	4 (2,2)	4	2	4	*	*	0	0	f		

CALL

4 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
CALL mn	11001101 ----n--- ----m--- (SP-1) = PCH; (SP-2) = PCL; PC = mn; SP = SP-2	13 (2,2,3,3,3)	11	7	9	-	-	-	-			
CALL (HL)	11101101 11101010 (SP-1) = PCH; (SP-2) = PCL; PC = HL; SP = SP-2	13 (2,2,3,3,3)	11	7	11	-	-	-	-			
CALL (IX)	11011101 11101010 (SP-1) = PCH; (SP-2) = PCL; PC = IX; SP = SP-2	13 (2,2,3,3,3)	11	7	11	-	-	-	-			
CALL (IY)	11111101 11101010 (SP-1) = PCH; (SP-2) = PCL; PC = IY; SP = SP-2	13 (2,2,3,3,3)	11	7	11	-	-	-	-			

CBM

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
CBM n	11101101 00000000 ----n--- tmp = [(HL) & ~n] [A & n]; (HL) = tmp; (DE) = tmp (only (DE) affected by IOI or IOE)	15 (2,2,2,1,2,3,3)	14	13	12	-	-	-	-			d

CCF

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
CCF	00111111 CF = ~CF	2	2	2	2	-	-	-	*	f	f	

CLR

2 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
CLR HL	10111111 HL = 0	2	2	2	2	-	-	-	-	r		
CLR HL	01111111 10111111 HL = 0	4 (2,2)	4	2	4	-	-	-	-	r		

CNVC

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
CNVC pp	11101101 10pp1110	8 (2,2,2,2)	8	6	8	-	-	-	W	f		
		convert pp to physical code address (ignore mbxcr bit-flip); set carry if write-protected address										

CNVD

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
CNVD pp	11101101 10pp1111	8 (2,2,2,2)	8	6	8	-	-	-	W	f		
convert pp to physical data address (ignore mbxcr bit-flip); set carry if write protected address												

CONVC

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
CONVC pp	11101101 00pp1110 convert pp to physical code address; set carry if write-protected address	8 (2,2,2,2)	8	6	8	-	-	-	W	f		

CONVD

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
CONVD pp	11101101 00pp1111 convert pp to physical data address; set carry if write protected address	8 (2,2,2,2)	8	6	8	-	-	-	W	f		

COPY

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
COPY	11101101 10000000	$7+7i$ (2,2,2,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			
(PY) = (PX); BC = BC-1; PY = PY+1; PX = PX+1; repeat while {BC != 0}												

COPYR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
COPYR	11101101 10001000 (PY) = (PX); BC = BC-1; PY = PY-1; PX = PX-1; repeat while {BC != 0}	$7+7i$ (2,2,2,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			

CP

22 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
CP (HL)	10111110 A - (HL)	5 (2,1,2)	5	5	5	*	*	V	*	f	r	s
CP (IX+d)	11011101 10111110 ----d--- A - (IX+d)	10 (2,2,2,2,2)	10	9	8	*	*	V	*	f	r	s
CP (IY+d)	11111101 10111110 ----d--- A - (IY+d)	10 (2,2,2,2,2)	10	9	8	*	*	V	*	f	r	s
CP n	11111110 ----n--- A - n	4 (2,2)	4	2	4	*	*	V	*	f	r	
CP r	10111-r- A - r	2	2	2	2	*	*	V	*	f	r	
CP HL,DE	11101101 01001000 HL - DE	4 (2,2)	4	2	4	*	*	V	*	f	r	
CP JKHL,BCDE	11101101 01011000 JKHL - BCDE	4 (2,2)	4	2	4	*	*	V	*	f	r	
CP HL,d	01001000 ----d--- HL - d (d sign-extended to 16 bits)	4 (2,2)	4	2	4	*	*	V	*	f	r	
CP (HL)	01111111 10111110 A - (HL)	7 (2,2,1,2)	7	5	7	*	*	V	*	f	r	s
CP r	01111111 10111-r- A - r	4 (2,2)	4	2	4	*	*	V	*	f	r	
CP HL,d	01111111 01001000 ----d--- HL - d (d sign-extended to 16 bits)	6 (2,2,2)	6	4	6	*	*	V	*	f	r	
CP HL, (IX+d)	11011101 10110001 ----d--- HL - (IX+d)	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	f	r	s
CP JKHL, (IX+d)	11011101 10110011 ----d--- JKHL - (IX+d)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	f	r	s
CP HL, (IY+d)	11111101 10110001 ----d--- HL - (IY+d)	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	f	r	s
CP JKHL, (IY+d)	11111101 10110011 ----d--- JKHL - (IY+d)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	f	r	s
CP JKHL,ps	01001001 011100ps JKHL - ps	4 (2,2)	4	2	4	*	*	V	*	f	r	
CP A, (ps+d)	01001001 011101ps ----d--- A - (ps+d)	10 (2,2,2,2,2)	10	9	8	*	*	V	*	f	r	
CP HL, (ps+d)	01001001 011110ps ----d--- HL - (ps+d)	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	f	r	
CP JKHL, (ps+d)	01001001 011111ps ----d--- JKHL - (ps+d)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	f	r	
CP A, (sp+n)	01001001 11111001 ----n--- A - (sp+n)	10 (2,2,2,2,2)	10	9	8	*	*	V	*	f	r	
CP HL, (sp+n)	01001001 11111010 ----n--- HL - (sp+n)	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	f	r	

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
CP JKHL, (sp+n)	01001001 11111011 ----n--- JKHL - (sp+n)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	f	r	

CPL

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
CPL	00101111 A = ~A	2	2	2	2	-	-	-	-	r	r	

DEC

9 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
DEC (HL)	00110101 (HL) = (HL) - 1	8 (2,1,2,3)	7	7	7	*	*	V	*	f		b
DEC (IX+d)	11011101 00110101 ----d--- (IX+d) = (IX+d) - 1	13 (2,2,2,2,2,3)	12	11	10	*	*	V	*	f		b
DEC (IY+d)	11111101 00110101 ----d--- (IY+d) = (IY+d) - 1	13 (2,2,2,2,2,3)	10	9	8	*	*	V	*	f		b
DEC IX	11011101 00101011 IX = IX - 1	4 (2,2)	4	2	4	-	-	-	-			
DEC IY	11111101 00101011 IY = IY - 1	4 (2,2)	4	2	4	-	-	-	-			
DEC r	00-r-101 r = r - 1	2	2	2	2	*	*	V	*	fr		r
DEC ss	00ss1011 ss = ss - 1	2	2	2	2	-	-	-	-	r		r
DEC (ps+d)	01001001 101101ps ----d--- (ps+d) = (ps+d) + 1	13 (2,2,2,2,2,3)	12	11	10	*	*	V	*	f		
DEC ps	01101101 ps001111 ps = ps - 1 (32-bit, independent of contents)	4 (2,2)	4	2	4	*	*	V	*	fr		r

DJNZ

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
DJNZ j	00010000 --(j-2)- B = B-1; if {B != 0} PC = PC + j	6 (2,2,2)	6	6	6	-	-	-	-	r	r	

DWJNZ

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
DWJNZ e	11101101 00010000 -(e-3)- BC = BC-1; if {BC != 0} PC = PC + e	8 (2,2,2,2)	8	6	8	-	-	-	-	r	r	

EX

16 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
EX (SP),HL	11101101 01010100 H <-> (SP+1); L <-> (SP)	15 (2,2,1,2,2,3,3)	13	7	13	-	-	-	-	r	r	
EX (SP),IX	11011101 11100011 IXH <-> (SP+1); IXL <-> (SP)	15 (2,2,1,2,2,3,3)	13	7	13	-	-	-	-			
EX (SP),IY	11111101 11100011 IYH <-> (SP+1); IYL <-> (SP)	15 (2,2,1,2,2,3,3)	13	7	13	-	-	-	-			
EX AF,AF'	00001000 AF <-> AF'	2	2	2	2	-	-	-	-			
EX DE,HL	11101011 if (!ALTD) then DE <-> HL else DE <-> HL'	2	2	2	2	-	-	-	-	s		
EX DE',HL	11100011 if (!ALTD) then DE' <-> HL else DE' <-> HL'	2	2	2	2	-	-	-	-	s		
EX BC',HL	11101101 01110100 if (!ALTD) then BC' <-> HL else BC' <-> HL'	4 (2,2)	4	2	4	-	-	-	-	s		
EX BC,HL	10110011 if (!ALTD) then BC <-> HL else BC <-> HL'	2	2	2	2	-	-	-	-	s		
EX JKHL,BCDE	10110100 if (!ALTD) then JKHL <-> BCDE else JKHL' <-> BCDE	2	2	2	2	-	-	-	-	s		
EX JK,HL	10111001 if (!ALTD) then JK <-> HL else JK <-> HL'	2	2	2	2	-	-	-	-	s		
EX JK',HL	11101101 01111100 if (!ALTD) then JK' <-> HL else JK' <-> HL'	4 (2,2)	4	2	4	-	-	-	-	s		
EX BC,HL	01111111 10110011 if (!ALTD) then BC <-> HL else BC <-> HL'	4 (2,2)	4	2	4	-	-	-	-	s		
EX JKHL,BCDE	01111111 10110100 if (!ALTD) then JKHL <-> BCDE else JKHL' <-> BCDE	4 (2,2)	4	2	4	-	-	-	-	s		
EX JK,HL	01111111 10111001 if (!ALTD) then JK <-> HL else JK <-> HL'	4 (2,2)	4	2	4	-	-	-	-	s		
EX JKHL,BCDE'	01000100 if (!ALTD) then JKHL <-> BCDE' else JKHL' <-> BCDE'	2	2	2	2	-	-	-	-	s		
EX JKHL,BCDE'	01111111 01000100 if (!ALTD) then JKHL <-> BCDE' else JKHL' <-> BCDE'	4 (2,2)	4	2	4	-	-	-	-	s		

EXP

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
EXP	11101101 11011001 PW <-> PW'; PX <-> PX'; PY <-> PY'; PZ <-> PZ'	4 (2,2)	4	2	4	-	-	-	-			

EXX

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
EXX	11011001 BC <-> BC'; DE <-> DE'; HL <-> HL'	2	2	2	2	-	-	-	-			

FLAG

4 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
FLAG cc,HL	11101101 101cx100 if (cc) then HL = 1 else HL = 0	4 (2,2)	4	2	4	-	-	-	-	r	f	
FLAG cc,HL	11101101 110cc100 if (cc) then HL = 1 else HL = 0	4 (2,2)	4	2	4	-	-	-	-	r	f	
FLAG cc,HL	11101101 111x0100 if (cc) then HL = 1 else HL = 0	4 (2,2)	4	2	4	-	-	-	-	r	f	
FLAG cc,HL	11101101 10011100 if (cc) then HL = 1 else HL = 0	4 (2,2)	4	2	4	-	-	-	-	r	f	

IBOX

2 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
IBOX A	11101101 00010010 A = inverse sbox(A)	4 (2,2)	4	2	4	-	-	-	-	r	r	
IBOX pp	01101101 ps011111 pp3 = ibox(pp3); pp2 = ibox(pp2); pp1 = ibox(pp1); pp0 = ibox(pp0)	4 (2,2)	4	2	4	-	-	-	-	r	r	

INC

9 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
INC (HL)	00110100 (HL) = (HL) + 1	8 (2,1,2,3)	7	7	7	*	*	V	*	f		b
INC (IX+d)	11011101 00110100 ----d--- (IX+d) = (IX+d) + 1	13 (2,2,2,2,2,3)	12	11	10	*	*	V	*	f		b
INC (IY+d)	11111101 00110100 ----d--- (IY+d) = (IY+d) + 1	13 (2,2,2,2,2,3)	10	9	8	*	*	V	*	f		b
INC IX	11011101 00100011 IX = IX + 1	4 (2,2)	4	2	4	-	-	-	-			
INC IY	11111101 00100011 IY = IY + 1	4 (2,2)	4	2	4	-	-	-	-			
INC r	00-r-100 r = r + 1	2	2	2	2	*	*	V	*	fr		r
INC ss	00ss0011 ss = ss + 1	2	2	2	2	-	-	-	-	r		r
INC (ps+d)	01001001 101001ps ----d--- (ps+d) = (ps+d) + 1	13 (2,2,2,2,2,3)	12	11	10	*	*	V	*	f		
INC ps	01101101 ps001101 ps = ps + 1 (32-bit, independent of contents)	4 (2,2)	4	2	4	*	*	V	*	fr		r

IOE

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
IOE	11011011 I/O external prefix	2	2	2	2	-	-	-	-			

I0I

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
I0I	11010011 I/O internal prefix	2	2	2	2	-	-	-	-			

IP

4 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
IP 0	11101101 01000110 IP = {IP[5:0], 00}	4 (2,2)	4	2	4	-	-	-	-			
IP 1	11101101 01010110 IP = {IP[5:0], 01}	4 (2,2)	4	2	4	-	-	-	-			
IP 2	11101101 01001110 IP = {IP[5:0], 10}	4 (2,2)	4	2	4	-	-	-	-			
IP 3	11101101 01011110 IP = {IP[5:0], 11}	4 (2,2)	4	2	4	-	-	-	-			

IPRES

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
IPRES	11101101 01011101 IP = {IP[1:0], IP[7:2]}	4 (2,2)	4	2	4	-	-	-	-			

JP

11 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
JP (HL)	11101001 PC = HL	4 (2,2)	4	4	4	-	-	-	-			
JP (IX)	11011101 11101001 PC = IX	6 (2,2,2)	6	4	6	-	-	-	-			
JP (IY)	11111101 11101001 PC = IY	6 (2,2,2)	6	4	6	-	-	-	-			
JP f,mn	11-f-010 ----n--- ----m--- if {f} PC = mn	7 (2,2,2,1)	7	5	5	-	-	-	-		f	
JP mn	11000011 ----n--- ----m--- PC = mn	7 (2,2,2,1)	7	5	5	-	-	-	-			
JP cc,mn	101cx010 ----n--- ----m--- if {cc} PC = mn	7 (2,2,2,1)	7	5	5	-	-	-	-		f	
JP cc,mn	01111111 101cx010 ----n--- ----m--- if {cc} PC = mn	9 (2,2,2,2,1)	9	5	5	-	-	-	-		f	
JP cc,mn	010x0011 ----n--- ----m--- if {cc} PC = mn	7 (2,2,2,1)	7	5	5	-	-	-	-		f	
JP cc,mn	01111111 010x0011 ----n--- ----m--- if {cc} PC = mn	9 (2,2,2,2,1)	9	5	5	-	-	-	-		f	
JP cc,mn	01001011 ----n--- ----m--- if {cc} PC = mn	7 (2,2,2,1)	7	5	5	-	-	-	-		f	
JP cc,mn	01111111 01001011 ----n--- ----m--- if {cc} PC = mn	9 (2,2,2,2,1)	9	5	5	-	-	-	-		f	

JR

8 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
JR cc,e	001cc000 --(e-2)- if {cc} PC = PC + e	6 (2,2,2)	5	4	5	-	-	-	-			f
JR e	00011000 --(e-2)- PC = PC + e	6 (2,2,2)	5	4	5	-	-	-	-			
JR cc,e	101cx000 -(e-2)- if {cc} PC = PC + e	6 (2,2,2)	5	4	5	-	-	-	-			f
JR cc,e	01111111 101cx000 -(e-2)- if {cc} PC = PC + e	6 (2,2,2)	8	6	8	-	-	-	-			f
JR cc,e	100x0000 -(e-2)- if {cc} PC = PC + e	6 (2,2,2)	5	4	5	-	-	-	-			f
JR cc,e	01111111 100x0000 -(e-2)- if {cc} PC = PC + e	6 (2,2,2)	8	6	8	-	-	-	-			f
JR cc,e	10001000 -(e-2)- if {cc} PC = PC + e	6 (2,2,2)	5	4	5	-	-	-	-			f
JR cc,e	01111111 10001000 -(e-2)- if {cc} PC = PC + e	6 (2,2,2)	8	6	8	-	-	-	-			f

JRE

6 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
JRE cc,ee	11101101 101cx011 -(ee-4)l- -(ee-4)h- if {cc} PC = PC + ee	10 (2,2,2,2,2)	10	6	6	-	-	-	-			f
JRE cc,ee	11101101 110cc011 -(ee-4)l- -(ee-4)h- if {cc} PC = PC + ee	10 (2,2,2,2,2)	10	6	6	-	-	-	-			f
JRE ee	10011000 -(ee-3)l- -(ee-3)h- PC = PC + ee	8 (2,2,2,2)	8	6	6	-	-	-	-			f
JRE ee	01111111 10011000 -(ee-3)l- -(ee-3)h- PC = PC + ee	10 (2,2,2,2,2)	10	6	6	-	-	-	-			f
JRE cc,ee	11101101 111x0011 -(ee-4)l- -(ee-4)h- if {cc} PC = PC + ee	10 (2,2,2,2,2)	10	6	6	-	-	-	-			f
JRE cc,ee	11101101 10011011 -(ee-4)l- -(ee-4)h- if {cc} PC = PC + ee	10 (2,2,2,2,2)	10	6	6	-	-	-	-			f

LCALL

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LCALL xpc,mn	11001111 ----n--- ----m--- --xpc---	20 (2,2,2,2,2,3,3,3,1)	17	11	15	-	-	-	-			
(SP-1) = XPCl; (SP-2) = PCH; (SP-3) = PCL; XPCl = xpc; XPCh = 0; PC = mn; SP = SP-3												

LD

166 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LD (BC),A	00000010 (BC) = A	7 (2,2,3)	6	6	6	-	-	-	-	r		d
LD (DE),A	00010010 (DE) = A	7 (2,2,3)	6	6	6	-	-	-	-	r		d
LD (HL),n	00110110 ----n--- (HL) = n	7 (2,2,3)	6	5	6	-	-	-	-			d
LD (HL),r	01110-r- (HL) = r	6 (2,1,3)	5	5	5	-	-	-	-	r		d
LD (HL+d),HL	11011101 11110100 ----d--- (HL+d) = L; (HL+d+1) = H	14 (2,2,2,2,3,3)	12	9	10	-	-	-	-	r		d
LD (IX+d),HL	11110100 ----d--- (IX+d) = L; (IX+d+1) = H	12 (2,2,2,3,3)	10	7	10	-	-	-	-	r		d
LD (IX+d),n	11011101 00110110 ----d--- ----n--- (IX+d) = n	12 (2,2,2,2,1,3)	12	8	8	-	-	-	-			d
LD (IX+d),r	11011101 01110-r- ----d--- (IX+d) = r	10 (2,2,2,1,3)	10	9	8	-	-	-	-	r		d
LD (IY+d),HL	11111101 11110100 ----d--- (IY+d) = L; (IY+d+1) = H	14 (2,2,2,2,3,3)	12	9	10	-	-	-	-	r		d
LD (IY+d),n	11111101 00110110 ----d--- ----n--- (IY+d) = n	12 (2,2,2,2,1,3)	12	8	8	-	-	-	-			d
LD (IY+d),r	11111101 01110-r- ----d--- (Iy+d) = r	11 (2,2,2,2,3)	10	9	8	-	-	-	-	r		d
LD (mn),A	00110010 ----n--- ----m--- (mn) = A	10 (2,2,2,1,3)	9	7	7	-	-	-	-	r		d
LD (mn),HL	00100010 ----n--- ----m--- (mn) = L; (mn+1) = H	13 (2,2,2,1,3,3)	11	7	9	-	-	-	-	r		d
LD (mn),IX	11011101 00100010 ----n--- ----m--- (mn) = IXL; (mn+1) = IXH	15 (2,2,2,2,1,3,3)	13	7	9	-	-	-	-			d
LD (mn),IY	11111101 00100010 ----n--- ----m--- (mn) = IYL; (mn+1) = IYH	15 (2,2,2,2,1,3,3)	13	7	9	-	-	-	-			d
LD (mn),ss	11101101 01ss0011 ----n--- ----m--- (mn) = ssl; (mn+1) = ssh	15 (2,2,2,2,1,3,3)	13	7	9	-	-	-	-	r		d
LD (SP+n),HL	11010100 ----n--- (SP+n) = L; (SP+n+1) = H	12 (2,2,2,3,3)	10	7	10	-	-	-	-	r		
LD (SP+n),IX	11011101 11010100 ----n--- (SP+n) = IXL; (SP+n+1) = IXH	14 (2,2,2,2,3,3)	12	9	10	-	-	-	-			
LD (SP+n),IY	11111101 11010100 ----n--- (SP+n) = IYL; (SP+n+1) = IYH	14 (2,2,2,2,3,3)	12	9	10	-	-	-	-			
LD A,(BC)	00001010 A = (BC)	6 (2,2,2)	6	6	6	-	-	-	-	r		s
LD A,(DE)	00011010 A = (DE)	6 (2,2,2)	6	6	6	-	-	-	-	r		s

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LD A, (mn)	00111010 ----n--- ----m--- A = (mn)	9 (2,2,2,1,2)	9	7	7	-	-	-	-	r		s
LD A,I	11101101 01010111 A = I	4 (2,2)	4	2	4	*	*	-	-	fr		
LD A,XPC	11101101 01110111 A = XPCl	4 (2,2)	4	2	4	-	-	-	-	r		
LD A,R	11101101 01011111 A = R	4 (2,2)	4	2	4	*	*	-	-	fr		
LD dd, (mn)	11101101 01dd1011 ----n--- ----m--- ddl = (mn); ddh = (mn+1)	13 (2,2,2,2,1,2,2)	13	7	9	-	-	-	-	r		s
LD dd',BC	11101101 01dd1001 dd' = BC (dd': 00-BC', 01-DE', 10-HL')	4 (2,2)	4	2	4	-	-	-	-		r	
LD dd',DE	11101101 01dd0001 dd' = DE (dd': 00-BC', 01-DE', 10-HL')	4 (2,2)	4	2	4	-	-	-	-		r	
LD dd,mn	00dd0001 ----n--- ----m--- dd = mn	6 (2,2,2)	6	4	4	-	-	-	-	r		
LD HL, (mn)	00101010 ----n--- ----m--- L = (mn); H = (mn+1)	11 (2,2,2,1,2,2)	11	7	9	-	-	-	-	r		s
LD HL, (HL+d)	11011101 11100100 ----d--- L = (HL+d); H = (HL+d+1)	12 (2,2,2,2,2,2)	12	9	10	-	-	-	-	r		s
LD HL, (IX+d)	11100100 ----d--- L = (IX+d); H = (IX+d+1)	10 (2,2,2,2,2,2)	10	7	10	-	-	-	-	r		s
LD HL, (IY+d)	11111101 11100100 ----d--- L = (IY+d); H = (IY+d+1)	12 (2,2,2,2,2,2)	12	9	10	-	-	-	-	r		s
LD HL, (SP+n)	11000100 ----n--- L = (SP+n); H = (SP+n+1)	10 (2,2,2,2,2,2)	10	7	10	-	-	-	-	r		
LD HL,IX	11011101 01111100 HL = IX	4 (2,2)	4	2	4	-	-	-	-	r		
LD HL,IY	11111101 01111100 HL = IY	4 (2,2)	4	2	4	-	-	-	-	r		
LD I,A	11101101 01000111 I = A	4 (2,2)	4	2	4	-	-	-	-		r	
LD IX, (mn)	11011101 00101010 ----n--- ----m--- IXL = (mn); IXH = (mn+1)	13 (2,2,2,2,1,2,2)	13	7	9	-	-	-	-			s
LD IX, (SP+n)	11011101 11000100 ----n--- IXL = (SP+n); IXH = (SP+n+1)	12 (2,2,2,2,2,2)	12	9	10	-	-	-	-			
LD IX,HL	11011101 01111101 IX = HL	4 (2,2)	4	2	4	-	-	-	-		r	
LD IX,mn	11011101 00100001 ----n--- ----m--- IX = mn	8 (2,2,2,2)	8	4	4	-	-	-	-			
LD IY, (mn)	11111101 00101010 ----n--- ----m--- IYL = (mn); IYH = (mn+1)	13 (2,2,2,2,1,2,2)	13	7	9	-	-	-	-			s
LD IY, (SP+n)	11111101 11000100 ----n--- IYL = (SP+n); IYH = (SP+n+1)	12 (2,2,2,2,2,2)	12	9	10	-	-	-	-			
LD IY,HL	11111101 01111101 IY = HL	4 (2,2)	4	2	4	-	-	-	-		r	

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LD IY,mn	11111101 00100001 ----n--- ----m--- IY = mn	8 (2,2,2,2)	8	4	4	-	-	-	-			
LD r,(HL)	01-r-110 r = (HL)	5 (2,1,2)	5	5	5	-	-	-	-	r		s
LD r,(IX+d)	11011101 01-r-110 ----d--- r = (IX+d)	10 (2,2,2,2,2)	10	9	8	-	-	-	-	r		s
LD r,(IY+d)	11111101 01-r-110 ----d--- r = (IY+d)	10 (2,2,2,2,2)	10	9	8	-	-	-	-	r		s
LD XPC,A	11101101 01100111 XPCL = A; XPCh = 0	4 (2,2)	4	2	4	-	-	-	-		r	
LD R,A	11101101 01001111 R = A	4 (2,2)	4	2	4	-	-	-	-		r	
LD r,n	00-r-110 ----n--- r = n	4 (2,2)	4	2	4	-	-	-	-	r		
LD r,r'	01-r--r' r = r'	2	2	2	2	-	-	-	-	r	r	
LD SP,HL	11111001 SP = HL	2	2	2	2	-	-	-	-		r	
LD SP,IX	11011101 11111001 SP = IX	4 (2,2)	4	2	4	-	-	-	-			
LD SP,IY	11111101 11111001 SP = IY	4 (2,2)	4	2	4	-	-	-	-			
LD pd,(SP+n)	11101101 00pd0100 ----n--- pd0 = (SP+n); pd1 = (SP+n+1); pd2 = (SP+n+2); pd3 = (SP+n+3)	16 (2,2,2,2,2,2,2,2)	16	11	14	-	-	-	-	r		
LD (SP+n),ps	11101101 00ps0101 ----n--- (SP+n) = ps0; (SP+n+1) = ps1; (SP+n+2) = ps2; (SP+n+3) = ps3	19 (2,2,2,1,3,3,3,3)	16	11	14	-	-	-	-		r	
LD HL,(ps+BC)	11101101 00ps0110 L = (ps+BC); H = (ps+BC+1)	11 (2,2,3,2,2)	11	7	11	-	-	-	-	r		
LD (pd+BC),HL	11101101 00pd0111 (pd+BC) = L; (pd+BC+1) = H	13 (2,2,3,3,3)	11	7	11	-	-	-	-		r	
LD pd,klmn	11101101 00pd1100 ----n--- ----m--- ----l--- ----k--- pd0 = n; pd1 = m; pd2 = l; pd3 = k	12 (2,2,2,2,2,2)	12	6	8	-	-	-	-	r		
LD HL,(SP+HL)	11101101 11111110 L = (SP+HL); H = (SP+HL+1)	11 (2,2,3,2,2)	11	7	11	-	-	-	-	r		
LD BCDE,(HL)	11011101 00011010 E = (HL); D = (HL+1); C = (HL+2); B = (HL+3)	14 (2,2,2,2,2,2,2)	14	8	14	-	-	-	-	r		s
LD (HL),BCDE	11011101 00011011 (HL) = E; (HL+1) = D; (HL+2) = C; (HL+3) = B	18 (2,2,2,3,3,3,3,3)	14	8	14	-	-	-	-		r	d
LD BCDE,(ps+HL)	11011101 00ps1100 E = (ps+HL); D = (ps+HL+1); C = (ps+HL+2); B = (ps+HL+3)	15 (2,2,3,2,2,2,2)	15	9	15	-	-	-	-	r		
LD (pd+HL),BCDE	11011101 00pd1101 (pd+HL) = E; (pd+HL+1) = D; (pd+HL+2) = C; (pd+HL+3) = B	19 (2,2,3,3,3,3,3,3)	15	9	15	-	-	-	-		r	
LD pd,BCDE	11011101 10pd1101 pd0 = E; pd1 = D; pd2 = C; pd3 = B	4 (2,2)	4	2	4	-	-	-	-	r	r	
LD BCDE,ps	11011101 11ps1101 E = ps0; D = ps1; C = ps2; B = ps3	4 (2,2)	4	2	4	-	-	-	-	r	r	

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LD BCDE, (ps+d)	11011101 00ps1110 ----d--- E = (ps+d); D = (ps+d+1); C = (ps+d+2); B = (ps+d+3)	16 (2,2,2,2,2,2,2,2)	16	11	14	-	-	-	-	r		
LD BCDE, (IX+d)	11011101 11001110 ----d--- E = (IX+d); D = (IX+d+1); C = (IX+d+2); B = (IX+d+3)	16 (2,2,2,2,2,2,2,2)	16	11	14	-	-	-	-	r		s
LD BCDE, (IY+d)	11011101 11011110 ----d--- E = (IY+d); D = (IY+d+1); C = (IY+d+2); B = (IY+d+3)	16 (2,2,2,2,2,2,2,2)	16	11	14	-	-	-	-	r		s
LD BCDE, (SP+n)	11011101 11101110 ----n--- E = (SP+n); D = (SP+n+1); C = (SP+n+2); B = (SP+n+3)	16 (2,2,2,2,2,2,2,2)	16	11	14	-	-	-	-	r		
LD BCDE, (SP+HL)	11011101 11111110 E = (SP+HL); D = (SP+HL+1); C = (SP+HL+2); B = (SP+HL+3)	15 (2,2,3,2,2,2,2,2)	15	9	15	-	-	-	-	r		
LD (pd+d), BCDE	11011101 00pd1111 ----d--- (pd+d) = E; (pd+d+1) = D; (pd+d+2) = C; (pd+d+3) = B	20 (2,2,2,2,3,3,3,3)	16	11	14	-	-	-	-		r	
LD (IX+d), BCDE	11011101 11001111 ----d--- (IX+d) = E; (IX+d+1) = D; (IX+d+2) = C; (IX+d+3) = B	20 (2,2,2,2,3,3,3,3)	16	11	14	-	-	-	-		r	d
LD (IY+d), BCDE	11011101 11011111 ----d--- (IY+d) = E; (IY+d+1) = D; (IY+d+2) = C; (IY+d+3) = B	20 (2,2,2,2,3,3,3,3)	16	11	14	-	-	-	-		r	d
LD (SP+n), BCDE	11011101 11101111 ----n--- (SP+n) = E; (SP+n+1) = D; (SP+n+2) = C; (SP+n+3) = B	20 (2,2,2,2,3,3,3,3)	16	11	14	-	-	-	-		r	
LD (SP+HL), BCDE	11011101 11111111 (SP+HL) = E; (SP+HL+1) = D; (SP+HL+2) = C; (SP+HL+3) = B	19 (2,2,3,3,3,3,3,3)	15	9	15	-	-	-	-		r	
LD JKHL, (HL)	11111101 00011010 L = (HL); H = (HL+1); K = (HL+2); J = (HL+3)	14 (2,2,2,2,2,2,2,2)	14	8	14	-	-	-	-	r		s
LD (HL), JKHL	11111101 00011011 (HL) = L; (HL+1) = H; (HL+2) = K; (HL+3) = J	18 (2,2,2,3,3,3,3,3)	14	8	14	-	-	-	-		r	d
LD JKHL, (ps+HL)	11111101 00ps1100 L = (ps+HL); H = (ps+HL+1); K = (ps+HL+2); J = (ps+HL+3)	15 (2,2,3,2,2,2,2,2)	15	9	15	-	-	-	-	r		
LD (pd+HL), JKHL	11111101 00pd1101 (pd+HL) = L; (pd+HL+1) = H; (pd+HL+2) = K; (pd+HL+3) = J	19 (2,2,3,3,3,3,3,3)	15	9	15	-	-	-	-		r	
LD pd, JKHL	11111101 10pd1101 pd0 = L; pd1 = H; pd2 = K; pd3 = J	4 (2,2)	4	2	4	-	-	-	-	r		r
LD JKHL, ps	11111101 11ps1101 L = ps0; H = ps1; K = ps2; J = ps3	4 (2,2)	4	2	4	-	-	-	-	r		r
LD JKHL, (ps+d)	11111101 00ps1110 ----d--- L = (ps+d); H = (ps+d+1); K = (ps+d+2); J = (ps+d+3)	16 (2,2,2,2,2,2,2,2)	16	11	14	-	-	-	-	r		
LD JKHL, (IX+d)	11111101 11001110 ----d--- L = (IX+d); H = (IX+d+1); K = (IX+d+2); J = (IX+d+3)	16 (2,2,2,2,2,2,2,2)	16	11	14	-	-	-	-	r		s
LD JKHL, (IY+d)	11111101 11011110 ----d--- L = (IY+d); H = (IY+d+1); K = (IY+d+2); J = (IY+d+3)	16 (2,2,2,2,2,2,2,2)	16	11	14	-	-	-	-	r		s
LD JKHL, (SP+n)	11111101 11101110 ----n--- L = (SP+n); H = (SP+n+1); K = (SP+n+2); J = (SP+n+3)	16 (2,2,2,2,2,2,2,2)	16	11	14	-	-	-	-	r		
LD JKHL, (SP+HL)	11111101 11111110 L = (SP+HL); H = (SP+HL+1); K = (SP+HL+2); J = (SP+HL+3)	15 (2,2,3,2,2,2,2,2)	15	9	15	-	-	-	-	r		
LD (pd+d), JKHL	11111101 00pd1111 ----d--- (pd+d) = L; (pd+d+1) = H; (pd+d+2) = K; (pd+d+3) = J	20 (2,2,2,2,3,3,3,3)	16	11	14	-	-	-	-		r	
LD (IX+d), JKHL	11111101 11001111 ----d--- (IX+d) = L; (IX+d+1) = H; (IX+d+2) = K; (IX+d+3) = J	20 (2,2,2,2,3,3,3,3)	16	11	14	-	-	-	-		r	d

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LD (IY+d),JKHL	11111101 11011111 ----d--- (IY+d) = L; (IY+d+1) = H; (IY+d+2) = K; (IY+d+3) = J	20 (2,2,2,2,3,3,3,3)	16	11	14	-	-	-	-	r		d
LD (SP+n),JKHL	11111101 11101111 ----n--- (SP+n) = L; (SP+n+1) = H; (SP+n+2) = K; (SP+n+3) = J	20 (2,2,2,2,3,3,3,3)	16	11	14	-	-	-	-	r		
LD (SP+HL),JKHL	11111101 11111111 (SP+HL) = L; (SP+HL+1) = H; (SP+HL+2) = K; (SP+HL+3) = J	19 (2,2,3,3,3,3,3)	15	9	15	-	-	-	-	r		
LD rr,(ps+d)	01101101 rrp0000 ----d--- rrl = (ps+d); rrh = (ps+d+1)	12 (2,2,2,2,2,2,2)	12	9	10	-	-	-	-	r		
LD (pd+d),rr	01101101 rrp0001 ----d--- (pd+d) = rrl; (pd+d+1) = rrh	14 (2,2,2,2,3,3)	12	9	10	-	-	-	-	r		
LD rr,(ps+HL)	01101101 rrp0010 rrl = (ps+HL); rrh = (ps+HL+1)	11 (2,2,3,2,2)	11	7	11	-	-	-	-	r		
LD (pd+HL),rr	01101101 rrp0011 (pd+HL) = rrl; (pd+HL+1) = rrh	13 (2,2,3,3,3)	11	7	11	-	-	-	-	r		
LD pd,(ps+d)	01101101 pdps1000 ----d--- pd0 = (ps+d); pd1 = (ps+d+1); pd2 = (ps+d+2); pd3 = (ps+d+3)	16 (2,2,2,2,2,2,2,2)	16	11	14	-	-	-	-	r		
LD (pd+d),ps	01101101 psp01001 ----d--- (pd+d) = ps0; (pd+d+1) = ps1; (pd+d+2) = ps2; (pd+d+3) = ps3	20 (2,2,2,2,3,3,3,3)	16	11	14	-	-	-	-	r		
LD pd,(ps+HL)	01101101 pdps1010 pd0 = (ps+HL); pd1 = (ps+HL+1); pd2 = (ps+HL+2); pd3 = (ps+HL+3)	15 (2,2,3,2,2,2,2)	15	9	15	-	-	-	-	r		
LD (pd+HL),ps	01101101 psp01011 (pd+HL) = ps0; (pd+HL+1) = ps1; (pd+HL+2) = ps2; (pd+HL+3) = ps3	19 (2,2,3,3,3,3,3)	15	9	15	-	-	-	-	r		
LD pd,ps+d	01101101 pdps1100 ----d--- pd = ps + d	6 (2,2,2)	6	4	4	-	-	-	-	r		
LD pd,ps+DE	01101101 pdps0110 pd = ps + DE	4 (2,2)	4	2	4	-	-	-	-	r		
LD pd,ps+HL	01101101 pdps1110 pd = ps + HL	4 (2,2)	4	2	4	-	-	-	-	r		
LD pd,ps+IX	01101101 pdps0100 pd = ps + IX	4 (2,2)	4	2	4	-	-	-	-	r		
LD pd,ps+IY	01101101 pdps0101 pd = ps + IY	4 (2,2)	4	2	4	-	-	-	-	r		
LD pd,ps	01101101 pdps0111 pd = ps	4 (2,2)	4	2	4	-	-	-	-	r		
LD HL,BC	10000001 HL = BC	2	2	2	2	-	-	-	-	r	r	
LD BC,HL	10010001 BC = HL	2	2	2	2	-	-	-	-	r	r	
LD HL,DE	10100001 HL = DE	2	2	2	2	-	-	-	-	r	r	
LD DE,HL	10110001 DE = HL	2	2	2	2	-	-	-	-	r	r	
LD (mn),BCDE	10000011 ----n--- ----m--- (mn) = E; (mn+1) = D; (mn+2) = C; (mn+3) = B	19 (2,2,2,1,3,3,3,3)	15	9	13	-	-	-	-	r		d
LD BCDE,(mn)	10010011 ----n--- ----m--- E = (mn); D = (mn+1); C = (mn+2); B = (mn+3)	15 (2,2,2,1,2,2,2,2)	15	9	13	-	-	-	-	r		s

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LD BCDE,d	10100011 ----d--- BCDE = d (d is sign-extended to 32 bits)	4 (2,2)	4	2	4	-	-	-	-	r		
LD (mn),JKHL	10000100 ----n--- ----m--- (mn) = L; (mn+1) = H; (mn+2) = K; (mn+3) = J	19 (2,2,2,1,3,3,3,3)	15	9	13	-	-	-	-	r		d
LD JKHL,(mn)	10010100 ----n--- ----m--- L = (mn); H = (mn+1); K = (mn+2); J = (mn+3)	15 (2,2,2,1,2,2,2,2)	15	9	13	-	-	-	-	r		s
LD JKHL,d	10100100 ----d--- JKHL = d (d is sign-extended to 32 bits)	4 (2,2)	4	2	4	-	-	-	-	r		
LD HL,(ps+d)	10ps0101 ----d--- L = (ps+d); H = (ps+d+1)	10 (2,2,2,2,2)	10	7	10	-	-	-	-	r		
LD (pd+d),HL	10pd0110 ----d--- (pd+d) = L; (pd+d+1) = H	12 (2,2,2,3,3)	10	7	10	-	-	-	-	r		
LD XPC,HL	10010111 XPCl = L; XPC _H = H	2	2	2	2	-	-	-	-	r		
LD (mn),JK	10001001 ----n--- ----m--- (mn) = K; (mn+1) = J	13 (2,2,2,1,3,3)	11	7	9	-	-	-	-	r		d
LD JK,(mn)	10011001 ----n--- ----m--- K = (mn); J = (mn+1)	11 (2,2,2,1,2,2)	11	7	9	-	-	-	-	r		s
LD JK,mn	10101001 ----n--- ----m--- K = n; J = m	6 (2,2,2)	6	4	4	-	-	-	-	r		
LD A,(ps+HL)	10ps1011 A = (ps+HL)	7 (2,3,2)	7	7	7	-	-	-	-	r		
LD (pd+HL),A	10pd1100 (pd+HL) = A	8 (2,3,3)	7	7	7	-	-	-	-	r		
LD A,(ps+d)	10ps1101 ----d--- A = (ps+d)	8 (2,2,2,2)	8	7	8	-	-	-	-	r		
LD (pd+d),A	10pd1110 ----d--- (pd+d) = A	9 (2,2,2,3)	8	7	8	-	-	-	-	r		
LD HL,XPC	10011111 HL = XPC	2	2	2	2	-	-	-	-	r		
LD A,r	01111rna A = r (only for r != A)	2	2	2	2	-	-	-	-	r		r
LD r,A	01rna111 r = A (only for r != A)	2	2	2	2	-	-	-	-	r		r
LD r,r'	01111111 01-r--r' r = r'	4 (2,2)	4	2	4	-	-	-	-	r		r
LD A,(IX+A)	11011101 00000110 A = (IX+A)	9 (2,2,3,2)	9	7	9	-	-	-	-	r		s
LD A,(IY+A)	11111101 00000110 A = (IY+A)	9 (2,2,3,2)	9	7	9	-	-	-	-	r		s
LD A,A	01101101 01111111 A = A	4 (2,2)	4	2	4	-	-	-	-	r		r
LD L,L	01101101 01101101 L = L	4 (2,2)	4	2	4	-	-	-	-	r		r
LD HTR,A	11101101 01000000 HTR = A	4 (2,2)	4	2	4	-	-	-	-	r		

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LD A,HTR	11101101 01010000 A = HTR	4 (2,2)	4	2	4	-	-	-	-	r		
LD pd, (HTR+HL)	11101101 00pd0001 pd0 = (HTR+HL); pd1 = (HTR+HL+1); pd2 = (HTR+HL+2); pd3 = (HTR+HL+3)	15 (2,2,3,2,2,2,2)	15	9	15	-	-	-	-			
LD HL,BC	01111111 10000001 HL = BC	2	4	2	4	-	-	-	-	r	r	
LD BC,HL	01111111 10010001 BC = HL	2	4	2	4	-	-	-	-	r	r	
LD HL,DE	01111111 10100001 HL = DE	2	4	2	4	-	-	-	-	r	r	
LD DE,HL	01111111 10110001 DE = HL	2	4	2	4	-	-	-	-	r	r	
LD (mn),BCDE	01111111 10000011 ----n--- ----m--- (mn) = E; (mn+1) = D; (mn+2) = C; (mn+3) = B	21 (2,2,2,2,1,3,3,3,3)	17	9	13	-	-	-	-		r	d
LD BCDE, (mn)	01111111 10010011 ----n--- ----m--- E = (mn); D = (mn+1); C = (mn+2); B = (mn+3)	17 (2,2,2,2,1,2,2,2,2)	17	9	13	-	-	-	-	r		s
LD BCDE,d	01111111 10100011 ----d--- BCDE = d (d is sign-extended to 32 bits)	6 (2,2,2)	6	4	4	-	-	-	-	r		
LD (mn),JKHL	01111111 10000100 ----n--- ----m--- (mn) = L; (mn+1) = H; (mn+2) = K; (mn+3) = J	21 (2,2,2,2,1,3,3,3,3)	17	9	13	-	-	-	-		r	d
LD JKHL, (mn)	01111111 10010100 ----n--- ----m--- L = (mn); H = (mn+1); K = (mn+2); J = (mn+3)	17 (2,2,2,2,1,2,2,2,2)	17	9	13	-	-	-	-	r		s
LD JKHL,d	01111111 10100100 ----d--- JKHL = d (d is sign-extended to 32 bits)	6 (2,2,2)	6	4	4	-	-	-	-	r		
LD HL, (ps+d)	01111111 10ps0101 ----d--- L = (ps+d); H = (ps+d+1)	12 (2,2,2,2,2,2)	12	9	10	-	-	-	-	r		
LD (pd+d),HL	01111111 10pd0110 ----d--- (pd+d) = L; (pd+d+1) = H	14 (2,2,2,2,3,3)	12	9	10	-	-	-	-		r	
LD XPC,HL	01111111 10010111 XPCL = L; XPCH = H	4 (2,2)	4	2	4	-	-	-	-		r	
LD (mn),JK	01111111 10001001 ----n--- ----m--- (mn) = K; (mn+1) = J	15 (2,2,2,2,1,3,3)	13	7	9	-	-	-	-		r	d
LD JK, (mn)	01111111 10011001 ----n--- ----m--- K = (mn); J = (mn+1)	13 (2,2,2,2,1,2,2)	13	7	9	-	-	-	-	r		s
LD JK,mn	01111111 10101001 ----n--- ----m--- K = n; J = m	8 (2,2,2,2)	8	4	4	-	-	-	-	r		
LD A, (ps+HL)	01111111 10ps1011 A = (ps+HL)	9 (2,2,3,2)	9	7	9	-	-	-	-	r		
LD (pd+HL),A	01111111 10pd1100 (pd+HL) = A	10 (2,2,3,3)	9	7	9	-	-	-	-		r	
LD A, (ps+d)	01111111 10ps1101 ----d--- A = (ps+d)	10 (2,2,2,2,2)	10	9	8	-	-	-	-	r		
LD (pd+d),A	01111111 10pd1110 ----d--- (pp+d) = A	11 (2,2,2,2,3)	10	9	8	-	-	-	-		r	

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LD HL,XPC	01111111 10011111 HL = XPC	4 (2,2)	4	2	4	-	-	-	-	r		
LD A,A	11001011 00110111 A = A	4 (2,2)	4	2	4	-	-	-	-	r	r	
LD B,B	11001011 00110000 B = B	4 (2,2)	4	2	4	-	-	-	-	r	r	
LD C,C	11001011 00110001 C = C	4 (2,2)	4	2	4	-	-	-	-	r	r	
LD D,D	11001011 00110010 D = D	4 (2,2)	4	2	4	-	-	-	-	r	r	
LD E,E	11001011 00110011 E = E	4 (2,2)	4	2	4	-	-	-	-	r	r	
LD H,H	11001011 00110100 H = H	4 (2,2)	4	2	4	-	-	-	-	r	r	
LD L,L	11001011 00110101 L = L	4 (2,2)	4	2	4	-	-	-	-	r	r	

LDD

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LDD	11101101 10101000 (DE) = (HL); BC = BC-1; DE = DE-1; HL = HL-1	10 (2,2,1,2,3)	9	7	9	-	-	*	-			d

LDDR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LDDR	11101101 10111000 (DE) = (HL); BC = BC-1; DE = DE-1; HL = HL-1; repeat while {BC != 0}	$\overset{6+7i}{(2,2,1,(2,3,2)i,1)}$	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			d

LDDSR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LDDSR	11101101 10011000 (DE) = (HL); BC = BC-1; HL = HL-1; repeat while {BC != 0}	$6+7i$ (2,2,1,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			d

LDF

16 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LDF <i>pd, (l_{mn})</i>	11101101 00pd1000 ----n--- ----m--- ----l--- pd0 = (l _{mn}); pd1 = (l _{mn} +1); pd2 = (l _{mn} +2); pd3 = (l _{mn} +3)	19 (2,2,2,2,2,1,2,2,2,2)	19	11	15	-	-	-	-	r		
LDF <i>(l_{mn}), ps</i>	11101101 00ps1001 ----n--- ----m--- ----l--- (l _{mn}) = ps0; (l _{mn} +1) = ps1; (l _{mn} +2) = ps2; (l _{mn} +3) = ps3	23 (2,2,2,2,2,1,3,3,3,3)	19	11	15	-	-	-	-		r	
LDF <i>rr, (l_{mn})</i>	11101101 00rr1010 ----n--- ----m--- ----l--- rrl = (l _{mn}); rrh = (l _{mn} +1)	15 (2,2,2,2,2,1,2,2)	13	7	9	-	-	-	-	r		
LDF <i>(l_{mn}), rr</i>	11101101 00rr1011 ----n--- ----m--- ----l--- (l _{mn}) = rrl; (l _{mn} +1) = rrh	17 (2,2,2,2,2,1,3,3)	15	9	11	-	-	-	-		r	
LDF <i>BCDE, (l_{mn})</i>	11011101 00001010 ----n--- ----m--- ----l--- E = (l _{mn}); D = (l _{mn} +1); C = (l _{mn} +2); B = (l _{mn} +3)	19 (2,2,2,2,2,1,2,2,2,2)	19	11	15	-	-	-	-	r		
LDF <i>(l_{mn}), BCDE</i>	11011101 00001011 ----n--- ----m--- ----l--- (l _{mn}) = E; (l _{mn} +1) = D; (l _{mn} +2) = C; (l _{mn} +3) = B	23 (2,2,2,2,2,1,3,3,3,3)	19	11	15	-	-	-	-		r	
LDF <i>JKHL, (l_{mn})</i>	11111101 00001010 ----n--- ----m--- ----l--- L = (l _{mn}); H = (l _{mn} +1); K = (l _{mn} +2); J = (l _{mn} +3)	19 (2,2,2,2,2,1,2,2,2,2)	19	11	15	-	-	-	-	r		
LDF <i>(l_{mn}), JKHL</i>	11111101 00001011 ----n--- ----m--- ----l--- (l _{mn}) = L; (l _{mn} +1) = H; (l _{mn} +2) = K; (l _{mn} +3) = J	23 (2,2,2,2,2,1,3,3,3,3)	19	11	15	-	-	-	-		r	
LDF <i>(l_{mn}), HL</i>	10000010 ----n--- ----m--- ----l--- (l _{mn}) = L; (l _{mn} +1) = H	15 (2,2,2,2,1,3,3)	13	7	11	-	-	-	-		r	
LDF <i>HL, (l_{mn})</i>	10010010 ----n--- ----m--- ----l--- L = (l _{mn}); H = (l _{mn} +1)	13 (2,2,2,2,1,2,2)	13	7	11	-	-	-	-	r		
LDF <i>(l_{mn}), A</i>	10001010 ----n--- ----m--- ----l--- (l _{mn}) = A	12 (2,2,2,2,1,3)	11	7	9	-	-	-	-		r	
LDF <i>A, (l_{mn})</i>	10011010 ----n--- ----m--- ----l--- A = (l _{mn})	11 (2,2,2,2,1,2)	11	7	9	-	-	-	-	r		
LDF <i>(l_{mn}), HL</i>	01111111 10000010 ----n--- ----m--- ----l--- (l _{mn}) = L; (l _{mn} +1) = H	17 (2,2,2,2,2,1,3,3)	15	9	11	-	-	-	-		r	
LDF <i>HL, (l_{mn})</i>	01111111 10010010 ----n--- ----m--- ----l--- L = (l _{mn}); H = (l _{mn} +1)	15 (2,2,2,2,2,1,2,2)	15	9	11	-	-	-	-	r		
LDF <i>(l_{mn}), A</i>	01111111 10001010 ----n--- ----m--- ----l--- (l _{mn}) = A	14 (2,2,2,2,2,1,3)	13	9	9	-	-	-	-		r	
LDF <i>A, (l_{mn})</i>	01111111 10011010 ----n--- ----m--- ----l--- A = (l _{mn})	13 (2,2,2,2,2,1,2)	13	9	9	-	-	-	-	r		

LDI

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LDI	11101101 10100000 (DE) = (HL); BC = BC-1; DE = DE+1; HL = HL+1	10 (2,2,1,2,3)	9	7	9	-	-	*	-			d

LDIR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LDIR	11101101 10110000 (DE) = (HL); BC = BC-1; DE = DE+1; HL = HL+1; repeat while {BC != 0}	$6+7i$ (2,2,1,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			d

LDISR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LDISR	11101101 10010000 (DE) = (HL); BC = BC-1; HL = HL+1; repeat while {BC != 0}	$6+7i$ (2,2,1,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			d

LDL

6 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LDL pd, (sp+n)	11101101 00pd0011 ----n--- pd0 = (SP+d); pd1 = (SP+d+1); pd2 = FF; pd3 = FF	13 (2,2,2,3,2,2)	12	9	10	-	-	-	-	r		
LDL pd, DE	11011101 10pd1111 pd = {FFFF, DE}	4 (2,2)	4	2	4	-	-	-	-	r	r	
LDL pd, HL	11111101 10pd1111 pd = {FFFF, HL}	4 (2,2)	4	2	4	-	-	-	-	r	r	
LDL pd, IX	11011101 10pd1100 pd = {FFFF, IX}	4 (2,2)	4	2	4	-	-	-	-	r		
LDL pd, IY	11111101 10pd1100 pd = {FFFF, IY}	4 (2,2)	4	2	4	-	-	-	-	r		
LDL pd, mn	11101101 00pd1101 ----n--- ----m--- pd = {FFFF, mn}	8 (2,2,2,2)	8	4	4	-	-	-	-	r		

LDP

12 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LDP (HL),HL	11101101 01100100 (HL) = L; (HL+1) = H. (Addr[19:16] = A[3:0])	12 (2,2,2,3,3)	10	6	10	-	-	-	-		r	
LDP (IX),HL	11011101 01100100 (IX) = L; (IX+1) = H. (Addr[19:16] = A[3:0])	12 (2,2,2,3,3)	10	6	10	-	-	-	-		r	
LDP (IY),HL	11111101 01100100 (IY) = L; (IY+1) = H. (Addr[19:16] = A[3:0])	12 (2,2,2,3,3)	10	6	10	-	-	-	-		r	
LDP (mn),HL	11101101 01100101 ----n--- ----m--- (mn) = L; (mn+1) = H. (Addr[19:16] = A[3:0])	15 (2,2,2,2,1,3,3)	13	7	9	-	-	-	-		r	
LDP (mn),IX	11011101 01100101 ----n--- ----m--- (mn) = IXL; (mn+1) = IXH. (Addr[19:16] = A[3:0])	15 (2,2,2,2,1,3,3)	13	7	9	-	-	-	-			
LDP (mn),IY	11111101 01100101 ----n--- ----m--- (mn) = IYL; (mn+1) = IYH. (Addr[19:16] = A[3:0])	15 (2,2,2,2,1,3,3)	13	7	9	-	-	-	-			
LDP HL,(HL)	11101101 01101100 L = (HL); H = (HL+1). (Addr[19:16] = A[3:0])	10 (2,2,2,2,2)	10	6	10	-	-	-	-		r	
LDP HL,(IX)	11011101 01101100 L = (IX); H = (IX+1). (Addr[19:16] = A[3:0])	10 (2,2,2,2,2)	10	6	10	-	-	-	-		r	
LDP HL,(IY)	11111101 01101100 L = (IY); H = (IY+1). (Addr[19:16] = A[3:0])	10 (2,2,2,2,2)	10	6	10	-	-	-	-		r	
LDP HL,(mn)	11101101 01101101 ----n--- ----m--- L = (mn); H = (mn+1). (Addr[19:16] = A[3:0])	13 (2,2,2,2,1,2,2)	13	7	9	-	-	-	-		r	
LDP IX,(mn)	11011101 01101101 ----n--- ----m--- IXL = (mn); IXH = (mn+1). (Addr[19:16] = A[3:0])	13 (2,2,2,2,1,2,2)	13	7	9	-	-	-	-			
LDP IY,(mn)	11111101 01101101 ----n--- ----m--- IYL = (mn); IYH = (mn+1). (Addr[19:16] = A[3:0])	13 (2,2,2,2,1,2,2)	13	7	9	-	-	-	-			

LJP

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LJP xpc,mn	11000111 ----n--- ----m--- --xpc-- XPCl = xpc; PC = mn	10 (2,2,2,2,2)	10	6	8	-	-	-	-			

LLCALL

3 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LLCALL l xpc,mn	10001111 ----n--- ----m--- ---xpl--- ---xph---	25 (2,2,2,2,2,2,3,3,3,1)	21	13	17	-	-	-	-			
	(SP-1) = XPCh; (SP-2) = XPCL; (SP-3) = PCH; (SP-4) = PCL; XPCL = xpl; XPCh = xph; PC = mn; SP = SP-4											
LLCALL (JKHL)	11101101 11111010	20 (2,2,3,3,3,3,1)	15	9	15	-	-	-	-			
	(SP-1) = XPCh; (SP-2) = XPCL; (SP-3) = PCH; (SP-4) = PCL; PC = HL; XPC = JK; SP = SP-4											
LLCALL l xpc,mn	01111111 10001111 ----n--- ----m--- ---xpl--- ---xph---	27 (2,2,2,2,2,2,2,3,3,3,1)	23	13	19	-	-	-	-			
	(SP-1) = XPCh; (SP-2) = XPCL; (SP-3) = PCH; (SP-4) = PCL; XPCL = xpl; XPCh = xph; PC = mn; SP = SP-4											

LLJP

6 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LLJP cc,lxpc,mn	11101101 101cx010 ----n--- ----m--- ---xpl-- ---xph-- if {cc} XPCL = xpl; XPCh = xph; PC = mn	14 (2,2,2,2,2,2,2)	14	8	10	-	-	-	-			f
LLJP cc,lxpc,mn	11101101 110cc010 ----n--- ----m--- ---xpl-- ---xph-- if {cc} XPCL = xpl; XPCh = xph; PC = mn	14 (2,2,2,2,2,2,2)	14	8	10	-	-	-	-			f
LLJP lxpc,mn	10000111 ----n--- ----m--- ---xpl--- ---xph--- XPCL = xpl; XPCh = xph; PC = mn	12 (2,2,2,2,2,2)	12	8	8	-	-	-	-			
LLJP lxpc,mn	01111111 10000111 ----n--- ----m--- ---xpl--- ---xph--- XPCL = xpl; XPCh = xph; PC = mn	14 (2,2,2,2,2,2,2)	14	8	10	-	-	-	-			
LLJP cc,lxpc,mn	11101101 111x0010 ----n--- ----m--- ---xpl-- ---xph-- if {cc} XPCL = xpl; XPCh = xph; PC = mn	14 (2,2,2,2,2,2,2)	14	8	10	-	-	-	-			f
LLJP cc,lxpc,mn	11101101 10011010 ----n--- ----m--- ---xpl-- ---xph-- if {cc} XPCL = xpl; XPCh = xph; PC = mn	14 (2,2,2,2,2,2,2)	14	8	10	-	-	-	-			f

LLRET

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LLRET	11101101 10001011 PCL = (SP); PCH = (SP+1); XPCL = (SP+2); XPCh = (SP+3); SP = SP+4	14 (2,2,2,2,2,2,2)	15	9	15	-	-	-	-			

LRET

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LRET	11101101 01000101 PCL = (SP); PCH = (SP+1); XPCL = (SP+2); XPCh = 0; SP = SP+3	13 (2,2,1,2,2,2,2)	13	9	13	-	-	-	-			

LSDDR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LSDDR	11101101 11011000 (DE) = (HL); BC = BC-1; DE = DE-1; repeat while {BC != 0}	$6+7i$ (2,2,1,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			S

LSDR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LSDR	11101101 11111000 (DE) = (HL); BC = BC-1; DE = DE-1; HL = HL-1; repeat while {BC != 0}	$\overset{6+7i}{(2,2,1,(2,3,2)i,1)}$	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			s

LSIDR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LSIDR	11101101 11010000 (DE) = (HL); BC = BC-1; DE = DE+1; repeat while {BC != 0}	$6+7i$ (2,2,1,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			s

LSIR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
LSIR	11101101 11110000 (DE) = (HL); BC = BC-1; DE = DE+1; HL = HL+1; repeat while {BC != 0}	$6+7i$ (2,2,1,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			s

MD5F1

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
MD5F1	01001001 11000011 JKHL + (PW & PX) ((~PW) & PY)	4 (2,2)	4	2	4	-	-	-	-			

MD5F2

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
MD5F2	01001001 11010011 JKHL = (PY & PW) ((~PY & PX)	4 (2,2)	4	2	4	-	-	-	-			

MD5F3

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
MD5F3	01001001 11100011 JKHL = PX ^ (PW (~PY))	4 (2,2)	4	2	4	-	-	-	-			

MUL

3 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
MUL	11110111 HL:BC = BC * DE	12 (2,10)	3	3	3	-	-	-	-			
MUL HL,DE	01011001 JK HL = HL * DE	12 (2,10)	12	12	12	-	-	-	-			
MUL HL,DE	01111111 01011001 JKHL = HL * DE	14 (2,2,10)	14	12	14	-	-	-	-			

MULU

4 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
MULU	10100111 HL:BC = BC * DE (unsigned)	12 (2,10)	12	12	12	-	-	-	-			
MULU	01111111 10100111 HL:BC = BC * DE (unsigned)	14 (2,2,10)	14	12	14	-	-	-	-			
MULU HL,DE	01101001 JKHL = HL * DE (unsigned)	12 (2,10)	12	12	12	-	-	-	-			
MULU HL,DE	01111111 01101001 JKHL = HL * DE (unsigned)	14 (2,2,10)	14	12	14	-	-	-	-			

NEG

5 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
NEG	11101101 01000100 A = 0 - A	4 (2,2)	4	2	4	*	*	V	*	fr	r	
NEG HL	01001101 HL = 0 - HL	2	2	2	2	*	*	V	*	fr	r	
NEG BCDE	11011101 01001101 BCDE = 0 - BCDE	4 (2,2)	4	2	4	*	*	V	*	fr	r	
NEG JKHL	11111101 01001101 JKHL = 0 - JKHL	4 (2,2)	4	2	4	*	*	V	*	fr	r	
NEG HL	01111111 01001101 HL = 0 - HL	4 (2,2)	4	2	4	*	*	V	*	fr	r	

NOP

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
NOP	00000000 No operation	2	2	2	2	-	-	-	-			

OR

23 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
OR (HL)	10110110 A = A (HL)	5 (2,1,2)	5	5	5	*	*	P	0	fr	r	s
OR (IX+d)	11011101 10110110 ----d--- A = A (IX+d)	10 (2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	s
OR (IY+d)	11111101 10110110 ----d--- A = A (IY+d)	10 (2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	s
OR HL,DE	11101100 HL = HL DE	2	2	2	2	*	*	P	0	fr	r	
OR IX,DE	11011101 11101100 IX = IX DE	4 (2,2)	4	2	4	*	*	P	0	f	r	
OR IY,DE	11111101 11101100 IY = IY DE	4 (2,2)	4	2	4	*	*	P	0	f	r	
OR n	11110110 ----n--- A = A n	4 (2,2)	4	2	4	*	*	P	0	fr	r	
OR r	10110-r- A = A r	2	2	2	2	*	*	P	0	fr	r	
OR (HL)	01111111 10110110 A = A (HL)	7 (2,2,1,2)	7	5	7	*	*	P	0	fr	r	s
OR A	10110111 A = A A	2	2	2	2	*	*	P	0	fr	r	
OR r	01111111 10110-r- A = A r	4 (2,2)	4	2	4	*	*	P	0	fr	r	
OR JKHL,BCDE	11101101 11110110 JKHL = JKHL BCDE	4 (2,2)	4	2	4	*	*	P	0	fr	r	
OR HL, (IX+d)	11011101 10110000 ----d--- HL = HL (IX+d)	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	s
OR JKHL, (IX+d)	11011101 10110010 ----d--- JKHL = JKHL (IX+d)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr	r	s
OR HL, (IY+d)	11111101 10110000 ----d--- HL = HL (IY+d)	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	s
OR JKHL, (IY+d)	11111101 10110010 ----d--- JKHL = JKHL (IY+d)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr	r	s
OR JKHL,ps	01001001 011000ps JKHL = JKHL ps	4 (2,2)	4	2	4	*	*	P	0	fr	r	
OR A, (ps+d)	01001001 011001ps ----d--- A = A (ps+d)	10 (2,2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	
OR HL, (ps+d)	01001001 011010ps ----d--- HL = HL (ps+d)	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	
OR JKHL, (ps+d)	01001001 011011ps ----d--- JKHL = JKHL (ps+d)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr	r	
OR A, (sp+n)	01001001 11101001 ----n--- A = A (sp+n)	10 (2,2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
OR HL, (sp+n)	01001001 11101010 ----n--- HL = HL (sp+n)	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	
OR JKHL, (sp+n)	01001001 11101011 ----n--- JKHL = JKHL (sp+n)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr	r	

PLDD

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PLDD	01001001 10101000 (PX) = (PY); BC = BC-1; PX = PX-1; PY = PY-1	11 (2,2,2,2,3)	10	7	10	-	-	*	-			d

PLDDR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PLDDR	01001001 10111000 (PX) = (PY); BC = BC-1; PX = PX-1; PY = PY-1; repeat while {BC != 0}	$\begin{matrix} 6+7i \\ (2,2,2,(2,3,2)i,1) \end{matrix}$	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			d

PLDDSR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PLDDSR	01001001 10011000 (PX) = (PY); BC = BC-1; PY = PY-1; repeat while {BC != 0}	$6+7i$ (2,2,2,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			d

PLDI

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PLDI	01001001 10100000 (PX) = (PY); BC = BC-1; PX = PX+1; PY = PY+1	11 (2,2,2,2,3)	10	7	10	-	-	*	-			d

PLDIR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PLDIR	01001001 10110000 (PX) = (PY); BC = BC-1; PX = PX+1; PY = PY+1; repeat while {BC != 0}	$\overset{6+7i}{(2,2,2,(2,3,2)i,1)}$	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			d

PLDISR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PLDISR	01001001 10010000 (PX) = (PY); BC = BC-1; PY = PY+1; repeat while {BC != 0}	$6+7i$ (2,2,2,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			d

PLSDDR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PLSDDR	01001001 11011000 (PX) = (PY); BC = BC-1; PX = PX-1; repeat while {BC != 0}	$6+7i$ (2,2,2,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			S

PLSDR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PLSDR	01001001 11111000 (PX) = (PY); BC = BC-1; PX = PX-1; PY = PY-1; repeat while {BC != 0}	$6+7i$ (2,2,2,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			s

PLSIDR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PLSIDR	01001001 11010000 (PX) = (PY); BC = BC-1; PX = PX+1; repeat while {BC != 0}	$6+7i$ (2,2,2,(2,3,2)i,1)	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			S

PLSIR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PLSIR	01001001 11110000 (PX) = (PY); BC = BC-1; PX = PX+1; PY = PY+1; repeat while {BC != 0}	$\begin{smallmatrix} 6+7i \\ (2,2,2,(2,3,2)i,1) \end{smallmatrix}$	7 + 6i	5 + 6i	7 + 6i	-	-	*	-			s

POP

8 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
POP IP	11101101 01111110 IP = (SP); SP = SP+1	7 (2,2,1,2)	7	5	7	-	-	-	-			
POP IX	11011101 11100001 IXL = (SP); IXLH = (SP+1); SP = SP+2	9 (2,2,1,2,2)	9	5	9	-	-	-	-			
POP IY	11111101 11100001 IYL = (SP); IYLH = (SP+1); SP = SP+2	9 (2,2,1,2,2)	9	5	9	-	-	-	-			
POP zz	11zz0001 zzl = (SP); zzh = (SP+1); SP = SP+2	7 (2,1,2,2)	7	5	7	-	-	-	-	r		
POP SU	11101101 01101110 SU = (SP); SP = SP+1	7 (2,2,1,2)	7	5	7	-	-	-	-			
POP pd	11101101 11pd0001 pd0 = (SP); pd1 = (SP+1); pd2 = (SP+2); pd3 = (SP+3); SP = SP+4	13 (2,2,1,2,2,2,2)	13	7	13	-	-	-	-	r		
POP BCDE	11011101 11110001 E = (SP); D = (SP+1); C = (SP+2); B = (SP+3); SP = SP+4	13 (2,2,1,2,2,2,2)	13	7	13	-	-	-	-	r		
POP JKHL	11111101 11110001 L = (SP); H = (SP+1); K = (SP+2); J = (SP+3); SP = SP+4	13 (2,2,1,2,2,2,2)	13	7	13	-	-	-	-	r		

PUMA

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PUMA	01001001 11000000 {CY:DE':(PZ)} = (PX) + [(PY) * DE + DE' + CY]; BC = BC-1; PX = PX+1; PY = PY+1; PZ = PZ+1; repeat while BC !=0	8+8i (2,2,2,(2,2,3,1)i,2)	7 + 8i	5 + 8i	7 + 8i	-	-	-	*			

PUMS

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PUMS	01001001 11001000 {CY:DE':(PZ)} = (PX) - [(PY) * DE + DE' + CY]; BC = BC-1; PX = PX+1; PY = PY+1; PZ = PZ+1; repeat while BC !=0	8+8i (2,2,2,(2,2,3,1)i,2)	7 + 8i	5 + 8i	7 + 8i	-	-	-	*			

PUSH

9 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
PUSH IP	11101101 01110110 (SP-1) = IP; SP = SP-1	10 (2,2,3,3)	9	7	9	-	-	-	-			
PUSH IX	11011101 11100101 (SP-1) = IXH; (SP-2) = IXL; SP = SP-2	13 (2,2,3,3,3)	11	7	11	-	-	-	-			
PUSH IY	11111101 11100101 (SP-1) = IYH; (SP-2) = IYL; SP = SP-2	13 (2,2,3,3,3)	11	7	11	-	-	-	-			
PUSH zz	11zz0101 (SP-1) = zzh; (SP-2) = zzl; SP = SP-2	11 (2,3,3,3)	9	7	9	-	-	-	-		r	
PUSH SU	11101101 01100110 (SP-1) = SU; SP = SP-1	10 (2,2,3,3)	9	7	9	-	-	-	-			
PUSH mn	11101101 10100101 ----n--- ----m--- (SP-1) = m; (SP-2) = n; SP = SP - 2	16 (2,2,2,2,2,3,3)	14	8	10	-	-	-	-			
PUSH ps	11101101 11ps0101 (SP-1) = ps3; (SP-2) = ps2; (SP-3) = ps1; (SP-4) = ps0; SP = SP-4	19 (2,2,3,3,3,3,3)	15	9	15	-	-	-	-		r	
PUSH BCDE	11011101 11110101 (SP-1) = B; (SP-2) = C; (SP-3) = D; (SP-4) = E; SP = SP-4	19 (2,2,3,3,3,3,3)	15	9	15	-	-	-	-		r	
PUSH JKHL	11111101 11110101 (SP-1) = J; (SP-2) = K; (SP-3) = H; (SP-4) = L; SP = SP-4	19 (2,2,3,3,3,3,3)	15	9	15	-	-	-	-		r	

RDMODE

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RDMODE	11101101 01111111 CY = SU[0]	4 (2,2)	4	2	4	-	-	-	*	f		

RES

4 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RES b, (HL)	11001011 10-b-110 (HL) = (HL) & ~bit	10 (2,2,1,2,3)	9	7	9	-	-	-	-			b
RES b, (IX+d)	11011101 11001011 ----d--- 10-b-110 (IX+d) = (IX+d) & ~bit	14 (2,2,2,2,1,2,3)	14	10	12	-	-	-	-			b
RES b, (IY+d)	11111101 11001011 ----d--- 10-b-110 (IY+d) = (IY+d) & ~bit	14 (2,2,2,2,1,2,3)	14	10	12	-	-	-	-			b
RES b, r	11001011 10-b--r- r = r & ~bit	4 (2,2)	4	2	4	-	-	-	-	r		r

RET

2 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RET	11001001 PCL = (SP); PCH = (SP+1); SP = SP+2	8 (2,1,2,2,1)	7	5	7	-	-	-	-			
RET f	11-f-000 if {f} PCL = (SP); PCH = (SP+1); SP = SP+2	2 8 (2,1,2,2,1)	2/8	2/6	2/8	-	-	-	-		f	

RETI

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RETI	11101101 01001101 IP = (SP); PCL = (SP+1); PCH = (SP+2); SP = SP+3	12 (2,2,1,2,2,2,1)	12	8	12	-	-	-	-			

RL

11 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RL (HL)	11001011 00010110 {CY, (HL)} = {(HL), CY}	10 (2,2,1,2,3)	9	7	9	*	*	P	*	f		b
RL (IX+d)	11011101 11001011 ----d--- 00010110 {CY, (IX+d)} = {(IX+d), CY}	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
RL (IY+d)	11111101 11001011 ----d--- 00010110 {CY, (IY+d)} = {(IY+d), CY}	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
RL DE	11110011 {CY, DE} = {DE, CY}	2	2	2	2	*	*	P	*	fr	fr	
RL r	11001011 00010-r- {CY, r} = {r, CY}	4 (2,2)	4	2	4	*	*	P	*	fr	fr	
RL b, BCDE	11011101 011010bb {CY, BCDE} = {BCDE, CY}; b=b-1; repeat while b!=0	4 (2,2)	4	2	4	*	*	P	*	fr	fr	
RL b, JKHL	11111101 011010bb {CY, JKHL} = {JKHL, CY}; b=b-1; repeat while b!=0; allowed b = {1,2,4}	4 (2,2)	4	2	4	*	*	P	*	fr	fr	
RL HL	01000010 {CY, HL} = {HL, CY}	2	2	2	2	*	*	P	*	fr	fr	
RL BC	01100010 {CY, BC} = {BC, CY}	2	2	2	2	*	*	P	*	fr	fr	
RL HL	01111111 01000010 {CY, HL} = {HL, CY}	4 (2,2)	4	2	4	*	*	P	*	fr	fr	
RL BC	01111111 01100010 {CY, BC} = {BC, CY}	4 (2,2)	4	2	4	*	*	P	*	fr	fr	

RL1REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RL1REG	01001001 10001101 JKHL = PW; PW = JKHL	4 (2,2)	4	2	4	-	-	-	-			

RL2REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RL2REG	01001001 10011101 JKHL = PW; PW = PX; PX = JKHL	4 (2,2)	4	2	4	-	-	-	-			

RL3REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RL3REG	01001001 10101101 JKHL = PW; PW = PX; PX = PY; PY = JKHL	4 (2,2)	4	2	4	-	-	-	-			

RL4REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RL4REG	01001001 10111101 JKHL = PW; PW = PX; PX = PY; PY = PZ; PZ = JKHL	4 (2,2)	4	2	4	-	-	-	-			

RL5REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RL5REG	01001001 11001101 JKHL = PW; PW = PX; PX = PY; PY = PZ; PZ = PW'; PW' = JKHL	4 (2,2)	4	2	4	-	-	-	-			

RL6REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RL6REG	01001001 11011101 JKHL = PW; PW = PX; PX = PY; PY = PZ; PZ = PW'; PW' = PX'; PX' = JKHL	4 (2,2)	4	2	4	-	-	-	-			

RL7REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RL7REG	01001001 11101101 JKHL = PW; PW = PX; PX = PY; PY = PZ; PZ = PW'; PW' = PX'; PX' = PY'; PY' = JKHL	4 (2,2)	4	2	4	-	-	-	-			

RL8REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RL8REG	01001001 11111101 JKHL = PW; PW = PX; PX = PY; PY = PZ; PZ = PW'; PW' = PX'; PX' = PY'; PY' = PZ'; PZ' JKHL	4 (2,2)	4	2	4	-	-	-	-			

RLA

3 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RLA	00010111 {CY,A} = {A,CY}	2	2	2	2	-	-	-	*	fr	fr	
RLA 8,BCDE	11011101 01101111 {A, BCDE} = {BCDE, A}	4 (2,2)	4	2	4	-	-	-	-	r	r	
RLA 8,JKHL	11111101 01101111 {A, JKHL} = {JKHL, A}	4 (2,2)	4	2	4	-	-	-	-	r	r	

RLC

12 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RLC (HL)	11001011 00000110 (HL) = {(HL)[6,0],(HL)[7]}; CY = (HL)[7]	10 (2,2,1,2,3)	9	7	9	*	*	P	*	f		b
RLC (IX+d)	11011101 11001011 ----d--- 00000110 (IX+d) = {(IX+d)[6,0],(IX+d)[7]}; CY = (IX+d)[7]	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
RLC (IY+d)	11111101 11001011 ----d--- 00000110 (IY+d) = {(IY+d)[6,0],(IY+d)[7]}; CY = (IY+d)[7]	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
RLC r	11001011 00000-r- r = {r[6,0],r[7]}; CY = r[7]	4 (2,2)	4	2	4	*	*	P	*	fr		r
RLC b,BCDE	11011101 010010bb BCDE = {BCDE[30,0],B[7]}; CY = B[7]; b=b-1; repeat while b!=0	4 (2,2)	4	2	4	*	*	P	*	fr		r
RLC b,JKHL	11111101 010010bb JKHL = {JKHL[30,0],J[7]}; CY = J[7]; b=b-1; repeat while b!=0; allowed b = {1,2,4}	4 (2,2)	4	2	4	*	*	P	*	fr		r
RLC DE	01010000 DE = {DE[14,0],D[7]}; CY = D[7]	2	2	2	2	*	*	P	*	fr		r
RLC BC	01100000 BC = {BC[14,0],B[7]}; CY = B[7]	2	2	2	2	*	*	P	*	fr		r
RLC 8,BCDE	11011101 01001111 BCDE = {CDE, B}	4 (2,2)	4	2	4	-	-	-	-	r		r
RLC 8,JKHL	11111101 01001111 JKHL = {KHL, J}	4 (2,2)	4	2	4	-	-	-	-	r		r
RLC DE	01111111 01010000 DE = {DE[14,0],D[7]}; CY = D[7]	2	4	2	4	*	*	P	*	fr		r
RLC BC	01111111 01100000 BC = {BC[14,0],B[7]}; CY = B[7]	2	4	2	4	*	*	P	*	fr		r

RLCA

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RLCA	00000111 A = {A[6,0],A[7]}; CY = A[7]	2	2	2	2	-	-	-	*	fr	r	

RR

12 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RR (HL)	11001011 00011110 {(HL),CY} = {CY,(HL)}	10 (2,2,1,2,3)	9	7	9	*	*	P	*	f		b
RR (IX+d)	11011101 11001011 ----d--- 00011110 {(IX+d),CY} = {CY,(IX+d)}	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
RR (IY+d)	11111101 11001011 ----d--- 00011110 {(IY+d),CY} = {CY,(IY+d)}	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
RR DE	11111011 {DE,CY} = {CY,DE}	2	2	2	2	*	*	P	*	fr	fr	
RR HL	11111100 {HL,CY} = {CY,HL}	2	2	2	2	*	*	P	*	fr	fr	
RR IX	11011101 11111100 {IX,CY} = {CY,IX}	4 (2,2)	4	2	4	*	*	P	*	f		
RR IY	11111101 11111100 {IY,CY} = {CY,IY}	4 (2,2)	4	2	4	*	*	P	*	f		
RR r	11001011 00011-r- {r,CY} = {CY,r}	4 (2,2)	4	2	4	*	*	P	*	fr	fr	
RR b,BCDE	11011101 011110bb {BCDE,CY} = {CY,BCDE}; b=b-1; repeat while b!=0	4 (2,2)	4	2	4	*	*	P	*	fr	fr	
RR b,JKHL	11111101 011110bb {JKHL,CY} = {CY,JKHL}; b=b-1; repeat while b!=0; allowed b = {1,2,4}	4 (2,2)	4	2	4	*	*	P	*	fr	fr	
RR BC	01100011 {BC,CY} = {CY,BC}	2	2	2	2	*	*	P	*	fr	fr	
RR BC	01111111 01100011 {BC,CY} = {CY,BC}	4 (2,2)	4	2	4	*	*	P	*	fr	fr	

RR1REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RR1REG	01001001 10001111 JKHL = PW; PW = JKHL	4 (2,2)	4	2	4	-	-	-	-			

RR2REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RR2REG	01001001 10011111 JKHL = PX; PW = JKHL; PX = PW	4 (2,2)	4	2	4	-	-	-	-			

RR3REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RR3REG	01001001 10101111 JKHL = PY; PW = JKHL; PX = PW; PY = PX	4 (2,2)	4	2	4	-	-	-	-			

RR4REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RR4REG	01001001 10111111 JKHL = PZ; PW = JKHL; PX = PW; PY = PX; PZ = PY	4 (2,2)	4	2	4	-	-	-	-			

RR5REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RR5REG	01001001 11001111 JKHL = PW'; PW = JKHL; PX = PW; PY = PX; PZ = PY; PW' = PZ	4 (2,2)	4	2	4	-	-	-	-			

RR6REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RR6REG	01001001 11011111 JKHL = PX'; PW = JKHL; PX = PW; PY = PX; PZ = PY; PW' = PZ; PX' = PW'	4 (2,2)	4	2	4	-	-	-	-			

RR7REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RR7REG	01001001 11101111 JKHL = PY'; PW = JKHL; PX = PW; PY = PX; PZ = PY; PW' = PZ; PX' = PW'; PY' = PX'	4 (2,2)	4	2	4	-	-	-	-			

RR8REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RR8REG	01001001 11111111 JKHL = PZ'; PW = JKHL; PX = PW; PY = PX; PZ = PY; PW' = PZ; PX' = PW'; PY' = PX'; PZ' = PY'	4 (2,2)	4	2	4	-	-	-	-			

RRA

3 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RRA	00011111 {A, CY} = {CY, A}	2	2	2	2	-	-	-	*	fr	fr	
RRA 8, BCDE	11011101 01111111 {A, BCDE} = {E, A, BCD}	4 (2,2)	4	2	4	-	-	-	-	r	r	
RRA 8, JKHL	11111101 01111111 {A, JKHL} = {L, A, JKH}	4 (2,2)	4	2	4	-	-	-	-	r	r	

RRC

12 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RRC (HL)	11001011 00001110 (HL) = {(HL)[0],(HL)[7,1]}; CY = (HL)[0]	10 (2,2,1,2,3)	9	7	9	*	*	P	*	f		b
RRC (IX+d)	11011101 11001011 ----d--- 00001110 (IX+d) = {(IX+d)[0],(IX+d)[7,1]}; CY = (IX+d)[0]	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
RRC (IY+d)	11111101 11001011 ----d--- 00001110 (IY+d) = {(IY+d)[0],(IY+d)[7,1]}; CY = (IY+d)[0]	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
RRC r	11001011 00001-r- r = {r[0],r[7,1]}; CY = r[0]	4 (2,2)	4	2	4	*	*	P	*	fr		r
RRC b,BCDE	11011101 010110bb BCDE = {B[7],BCDE[31,1]}; CY = E[0]	4 (2,2)	4	2	4	*	*	P	*	fr		r
RRC b,JKHL	11111101 010110bb JKHL = {J[7],JKHL[31,1]}; CY = L[0]; b=b-1; repeat while b!=0; allowed b = {1,2,4}	4 (2,2)	4	2	4	*	*	P	*	fr		r
RRC DE	01010001 DE = {D[7],DE[15,1]}; CY = E[0]	2	2	2	2	*	*	P	*	fr		r
RRC BC	01100001 BC = {B[7],BC[15,1]}; CY = C[0]	2	2	2	2	*	*	P	*	fr		r
RRC 8,BCDE	11011101 01011111 BCDE = {E, BCD}	4 (2,2)	4	2	4	-	-	-	-	r		r
RRC 8,JKHL	11111101 01011111 JKHL = {L, JKH}	4 (2,2)	4	2	4	-	-	-	-	r		r
RRC DE	01111111 01010001 DE = {D[7],DE[15,1]}; CY = E[0]	2	4	2	4	*	*	P	*	fr		r
RRC BC	01111111 01100001 BC = {B[7],BC[15,1]}; CY = C[0]	2	4	2	4	*	*	P	*	fr		r

RRCA

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RRCA	00001111 A = {A[0],A[7,1]}; CY = A[0]	2	2	2	2	-	-	-	*	fr	r	

RST

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
RST v	11-v-111 (SP-1) = PCH; (SP-2) = PCL; SP = SP - 2; PC = {R, 0, v, 0000}; SU <= {SU[5:0], 00}	11 (2,3,3,3)	9	7	9	-	-	-	-			

SBC

20 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SBC (IX+d)	11011101 10011110 ----d--- A = A - (IX+d) - CY	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	fr	s
SBC (IY+d)	11111101 10011110 ----d--- A = A - (IY+d) - CY	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	fr	s
SBC A, (HL)	10011110 A = A - (HL) - CY	5 (2,1,2)	5	5	5	*	*	V	*	fr	fr	s
SBC A, n	11011110 ----n--- A = A - n - CY	4 (2,2)	4	2	4	*	*	V	*	fr	fr	
SBC A, r	10011-r- A = A - r - CY	2	2	2	2	*	*	V	*	fr	fr	
SBC HL, ss	11101101 01ss0010 HL = HL - ss - CF	4 (2,2)	4	2	4	*	*	V	*	fr	fr	
SBC A, (HL)	01111111 10011110 A = A - (HL) - CY	7 (2,2,1,2)	7	5	7	*	*	V	*	fr	f	s
SBC A, r	01111111 10011-r- A = A - r - CY	4 (2,2)	4	2	4	*	*	V	*	fr	fr	
SBC JKHL, BCDE	11101101 11011110 JKHL = JKHL - BCDE - CY	4 (2,2)	4	2	4	*	*	V	*	fr	fr	
SBC HL, (IX+d)	11011101 10010001 ----d--- HL = HL - (IX+d) - CY	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	fr	s
SBC JKHL, (IX+d)	11011101 10010011 ----d--- JKHL = JKHL - (IX+d) - CY	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	fr	s
SBC HL, (IY+d)	11111101 10010001 ----d--- HL = HL - (IY+d) - CY	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	fr	s
SBC JKHL, (IY+d)	11111101 10010011 ----d--- JKHL = JKHL - (IY+d) - CY	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	fr	s
SBC JKHL, ps	01001001 001100ps JKHL = JKHL - ps - CY	4 (2,2)	4	2	4	*	*	V	*	fr	fr	
SBC A, (ps+d)	01001001 001101ps ----d--- A = A - (ps+d) - CY	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	fr	
SBC HL, (ps+d)	01001001 001110ps ----d--- HL = HL - (ps+d) - CY	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	fr	
SBC JKHL, (ps+d)	01001001 001111ps ----d--- JKHL = JKHL - (ps+d) - CY	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	fr	
SBC A, (sp+n)	01001001 10111001 ----n--- A = A - (sp+n) - CY	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	fr	
SBC HL, (sp+n)	01001001 10111010 ----n--- HL = HL - (sp+n) - CY	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	fr	
SBC JKHL, (sp+n)	01001001 10111011 ----n--- JKHL = JKHL - (sp+n) - CY	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	fr	

SBOX

2 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SBOX A	11101101 00000010 A = sbox(A)	4 (2,2)	4	2	4	-	-	-	-	r	r	
SBOX pp	01101101 ps011101 pp3 = sbox(pp3); pp2 = sbox(pp2); pp1 = sbox(pp1); pp0 = sbox(pp0)	4 (2,2)	4	2	4	-	-	-	-	r	r	

SCALL

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SCALL	11101101 01010101 (SP-1) = PCH; (SP-2) = PCL; (SP-3) = SU; SP = SP - 3; PC = {IIR, 01100000}; SU = {SU[5:0], 00}	16 (2,2,3,3,3,3)	13	9	13	-	-	-	-			

SCF

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SCF	00110111 CF = 1	2	2	2	2	-	-	-	1	f		

SET

4 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SET b, (HL)	11001011 11-b-110 (HL) = (HL) bit	10 (2,2,1,2,3)	9	7	9	-	-	-	-			b
SET b, (IX+d)	11011101 11001011 ----d--- 11-b-110 (IX+d) = (IX+d) bit	14 (2,2,2,2,1,2,3)	14	10	12	-	-	-	-			b
SET b, (IY+d)	11111101 11001011 ----d--- 11-b-110 (IY+d) = (IY+d) bit	14 (2,2,2,2,1,2,3)	14	10	12	-	-	-	-			b
SET b, r	11001011 11-b--r- r = r bit	4 (2,2)	4	2	4	-	-	-	-	r		r

SETSYSP

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SETSYSP mn	11101101 10110001 ----n--- ----m--- SU = {SU[1:0], SU[7:2]}; tmp _l = (SP); tmp _h = (SP+1); SP = SP+2; if {tmp != mn} System Violation	12 (2,2,2,2,2,2)	12	6	8	-	-	-	-			

SETUSR

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SETUSR	11101101 01101111 SU = {SU[5:0], 01}	4 (2,2)	4	2	4	-	-	-	-			

SETUSRP

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SETUSRP mn	11101101 10110101 ----n--- ----m--- SU = {SU[7:2], 01}, (SP-1) = m; (SP-2) = n; SP = SP-2	16 (2,2,2,2,2,3,3)	14	8	10	-	-	-	-			

SHAF1

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SHAF1	01001001 11000010 JKHL = ((PX ^ PY) & PW) ^ PY	4 (2,2)	4	2	4	-	-	-	-			

SHAF2

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SHAF2	01001001 11010010 JKHL = ((PW PX) & PY) (PW & PX)	4 (2,2)	4	2	4	-	-	-	-			

SHAF3

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SHAF3	01001001 11100010 JKHL = PW ^ PX ^ PY	4 (2,2)	4	2	4	-	-	-	-			

SL1REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SL1REG	01001001 10001100 JKHL = PW	4 (2,2)	4	2	4	-	-	-	-			

SL2REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SL2REG	01001001 10011100 JKHL = PW; PW = PX	4 (2,2)	4	2	4	-	-	-	-			

SL3REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SL3REG	01001001 10101100 JKHL = PW; PW = PX; PX = PY	4 (2,2)	4	2	4	-	-	-	-			

SL4REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SL4REG	01001001 10111100 JKHL = PW; PW = PX; PX = PY; PY = PZ	4 (2,2)	4	2	4	-	-	-	-			

SL5REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SL5REG	01001001 11001100 JKHL = PW; PW = PX; PX = PY; PY = PZ; PZ = PW'	4 (2,2)	4	2	4	-	-	-	-			

SL6REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SL6REG	01001001 11011100 JKHL = PW; PW = PX; PX = PY; PY = PZ; PZ = PW'; PW' = PX'	4 (2,2)	4	2	4	-	-	-	-			

SL7REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SL7REG	01001001 11101100 JKHL = PW; PW = PX; PX = PY; PY = PZ; PZ = PW'; PW' = PX'; PX' = PY'	4 (2,2)	4	2	4	-	-	-	-			

SL8REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SL8REG	01001001 11111100 JKHL = PW; PW = PX; PX = PY; PY = PZ; PZ = PW'; PW' = PX'; PX' = PY'; PY' = PZ'	4 (2,2)	4	2	4	-	-	-	-			

SLA

6 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SLA (HL)	11001011 00100110 (HL) = {(HL)[6,0],0}; CY = (HL)[7]	10 (2,2,1,2,3)	9	7	9	*	*	P	*	f		b
SLA (IX+d)	11011101 11001011 ----d--- 00100110 (IX+d) = {(IX+d)[6,0],0}; CY = (IX+d)[7]	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
SLA (IY+d)	11111101 11001011 ----d--- 00100110 (IY+d) = {(IY+d)[6,0],0}; CY = (IY+d)[7]	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
SLA r	11001011 00100-r- r = {r[6,0],0}; CY = r[7]	4 (2,2)	4	2	4	*	*	P	*	fr		r
SLA b,BCDE	11011101 100010bb BCDE = {BCDE[30,0],0}; CY = B[7]	4 (2,2)	4	2	4	*	*	P	*	fr		r
SLA b,JKHL	11111101 100010bb JKHL = {JKHL[30,0],0}; CY = J[7]; b=b-1; repeat while b!=0; allowed b = {1,2,4}	4 (2,2)	4	2	4	*	*	P	*	fr		r

SLL

2 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SLL b,BCDE	11011101 101010bb BCDE = {BCDE[30,0],0}; CY = B[7]	4 (2,2)	4	2	4	*	*	P	*	fr	r	
SLL b,JKHL	11111101 101010bb JKHL = {JKHL[30,0],0}; CY = J[7]; b=b-1; repeat while b!=0; allowed b = {1,2,4}	4 (2,2)	4	2	4	*	*	P	*	fr	r	

SR1REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SR1REG	01001001 10001110 JKHL = PW	4 (2,2)	4	2	4	-	-	-	-			

SR2REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SR2REG	01001001 10011110 JKHL = PX; PX = PW	4 (2,2)	4	2	4	-	-	-	-			

SR3REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SR3REG	01001001 10101110 JKHL = PY; PX = PW; PY = PX	4 (2,2)	4	2	4	-	-	-	-			

SR4REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SR4REG	01001001 10111110 JKHL = PZ; PX = PW; PY = PX; PZ = PY	4 (2,2)	4	2	4	-	-	-	-			

SR5REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SR5REG	01001001 11001110 JKHL = PW' ; PX = PW; PY = PX; PZ = PY; PW' = PZ	4 (2,2)	4	2	4	-	-	-	-			

SR6REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SR6REG	01001001 11011110 JKHL = PX' ; PX = PW; PY = PX; PZ = PY; PW' = PZ; PX' = PW'	4 (2,2)	4	2	4	-	-	-	-			

SR7REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SR7REG	01001001 11101110 JKHL = PY'; PX = PW; PY = PX; PZ = PY; PW' = PZ; PX' = PW'; PY' = PX'	4 (2,2)	4	2	4	-	-	-	-			

SR8REG

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SR8REG	01001001 11111110 JKHL = PZ'; PX = PW; PY = PX; PZ = PY; PW' = PZ; PX' = PW'; PY' = PX'; PZ' = PY'	4 (2,2)	4	2	4	-	-	-	-			

SRA

6 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SRA (HL)	11001011 00101110 (HL) = {(HL)[7],(HL)[7,1]}; CY = (HL)[0]	10 (2,2,1,2,3)	9	7	9	*	*	P	*	f		b
SRA (IX+d)	11011101 11001011 ----d--- 00101110 (IX+d) = {(IX+d)[7],(IX+d)[7,1]}; CY = (IX+d)[0]	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
SRA (IY+d)	11111101 11001011 ----d--- 00101110 (IY+d) = {(IY+d)[7],(IY+d)[7,1]}; CY = (IY+d)[0]	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
SRA r	11001011 00101-r- r = {r[7],r[7,1]}; CY = r[0]	4 (2,2)	4	2	4	*	*	P	*	fr		r
SRA b,BCDE	11011101 100110bb BCDE = {B[7],BCDE[31,1]}; CY = E[0]	4 (2,2)	4	2	4	*	*	P	*	fr		r
SRA b,JKHL	11111101 100110bb JKHL = {J[7],JKHL[31,1]}; CY = L[0]; b=b-1; repeat while b!=0; allowed b = {1,2,4}	4 (2,2)	4	2	4	*	*	P	*	fr		r

SRET

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SRET	11101101 10000011 SU = (SP); PCL = (SP+1); PCH = (SP+2); SP = SP+3	12 (2,2,1,2,2,2,1)	12	8	12	-	-	-	-			

SRL

6 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SRL (HL)	11001011 00111110 (HL) = {0, (HL)[7,1]}; CY = (HL)[0]	10 (2,2,1,2,3)	9	7	9	*	*	P	*	f		b
SRL (IX+d)	11011101 11001011 ----d--- 00111110 (IX+d) = {0, (IX+d)[7,1]}; CY = (IX+d)[0]	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
SRL (IY+d)	11111101 11001011 ----d--- 00111110 (IY+d) = {0, (IY+d)[7,1]}; CY = (IY+d)[0]	14 (2,2,2,2,1,2,3)	14	10	12	*	*	P	*	f		b
SRL r	11001011 00111-r- r = {0, r[7,1]}; CY = r[0]	4 (2,2)	4	2	4	*	*	P	*	fr		r
SRL b, BCDE	11011101 101110bb BCDE = {0, BCDE[31,1]}; CY = E[0]	4 (2,2)	4	2	4	*	*	P	*	fr		r
SRL b, JKHL	11111101 101110bb JKHL = {0, JKHL[31,1]}; CY = L[0]; b=b-1; repeat while b!=0; allowed b = {1,2,4}	4 (2,2)	4	2	4	*	*	P	*	fr		r

SUB

23 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SUB (HL)	10010110 A = A - (HL)	5 (2,1,2)	5	5	5	*	*	V	*	fr	r	s
SUB (IX+d)	11011101 10010110 ----d---	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	r	s
SUB (IY+d)	11111101 10010110 ----d---	10 (2,2,2,2,2)	10	9	8	*	*	V	*	fr	r	s
SUB n	11010110 ----n---	4 (2,2)	4	2	4	*	*	V	*	fr	r	
SUB r	10010-r- A = A - r	2	2	2	2	*	*	V	*	fr	r	
SUB HL, JK	01000101 HL = HL - JK	2	2	2	2	-	-	-	*	fr	r	
SUB HL, DE	01010101 HL = HL - DE	2	2	2	2	-	-	-	*	fr	r	
SUB (HL)	01111111 10010110 A = A - (HL)	7 (2,2,1,2)	7	5	7	*	*	V	*	fr	r	s
SUB r	01111111 10010-r- A = A - r	4 (2,2)	4	2	4	*	*	V	*	fr	r	
SUB JKHL, BCDE	11101101 11010110 JKHL = JKHL - BCDE	4 (2,2)	4	2	4	-	-	-	*	fr	r	
SUB HL, JK	01111111 01000101 HL = HL - JK	4 (2,2)	4	2	4	-	-	-	*	fr	r	
SUB HL, DE	01111111 01010101 HL = HL - DE	4 (2,2)	4	2	4	-	-	-	*	fr	r	
SUB HL, (IX+d)	11011101 10010000 ----d---	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	r	s
SUB JKHL, (IX+d)	11011101 10010010 ----d---	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	r	s
SUB HL, (IY+d)	11111101 10010000 ----d---	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	r	s
SUB JKHL, (IY+d)	11111101 10010010 ----d---	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	r	s
SUB JKHL, ps	01001001 001000ps JKHL = JKHL - ps	4 (2,2)	4	2	4	*	*	V	*	fr	r	
SUB A, (ps+d)	01001001 001001ps ----d---	10 (2,2,2,2,2,2)	10	9	8	*	*	V	*	fr	r	
SUB HL, (ps+d)	01001001 001010ps ----d---	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	r	
SUB JKHL, (ps+d)	01001001 001011ps ----d---	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	r	
SUB A, (sp+n)	01001001 10101001 ----n---	10 (2,2,2,2,2,2)	10	9	8	*	*	V	*	fr	r	

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SUB HL, (sp+n)	01001001 10101010 ----n--- HL = HL - (sp+n)	12 (2,2,2,2,2,2)	12	9	10	*	*	V	*	fr	r	
SUB JKHL, (sp+n)	01001001 10101011 ----n--- JKHL = JKHL - (sp+n)	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	V	*	fr	r	

SURES

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SURES	11101101 01111101 SU = {SU[1:0], SU[7:2]}	4 (2,2)	4	2	4	-	-	-	-			

SWAP

7 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SWAP r	11101101 1-r-0111 r[7:0] = r[0:7]	4 (2,2)	4	2	4	-	-	-	-	r	r	
SWAP BC	11101101 11001111 B = C; C = B	4 (2,2)	4	2	4	-	-	-	-	r	r	
SWAP DE	11101101 11011111 D = E; E = D	4 (2,2)	4	2	4	-	-	-	-	r	r	
SWAP HL	11101101 11101111 H = L; L = H	4 (2,2)	4	2	4	-	-	-	-	r	r	
SWAP JK	11101101 11111111 J = K; K = J	4 (2,2)	4	2	4	-	-	-	-	r	r	
SWAP BCDE	11011101 00110010 B = E; C = D; D = C; E = B	4 (2,2)	4	2	4	-	-	-	-	r	r	
SWAP JKHL	11111101 00110010 J = L; K = H; H = K; L = J	4 (2,2)	4	2	4	-	-	-	-	r	r	

SYSCALL

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
SYSCALL	11101101 01110101 (SP-1) = PCH; (SP-2) = PCL; SP = SP - 2; PC = {R, 01100000}; SU = {SU[5:0], 00}	13 (2,2,3,3,3)	11	7	11	-	-	-	-			

TEST

8 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
TEST BC	11101101 01001100 BC 0	4 (2,2)	4	2	4	*	*	P	0	f	r	
TEST IX	11011101 01001100 IX 0	4 (2,2)	4	2	4	*	*	P	0	f		
TEST BCDE	11011101 01011100 BCDE 0	4 (2,2)	4	2	4	*	*	P	0	f	r	
TEST IY	11111101 01001100 IY 0	4 (2,2)	4	2	4	*	*	P	0	f		
TEST JKHL	11111101 01011100 JKHL 0	4 (2,2)	4	2	4	*	*	P	0	f	r	
TEST HL	01001100 HL 0	2	2	2	2	*	*	P	0	f	r	
TEST HL	01111111 01001100 HL 0	4 (2,2)	4	2	4	*	*	P	0	f	r	
TEST DE	11101101 01011100 DE 0	4 (2,2)	4	2	4	*	*	P	0	f	r	

TSTNULL

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
TSTNULL pp	11101101 10pp0110 Z = 1 if ((pp == FFFF0000) (pp == 00000000)); C = 1 if (pp == FFFF0000)	4 (2,2)	4	2	4	-	*	-	*	f	r	

UMA

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
UMA	11101101 11000000 {CY:DE':(HL)} = (IX) + [(IY) * DE + DE' + CY]; BC = BC-1; IX = IX+1; IY = IY+1; HL = HL+1; repeat while BC !=0	8+8i (2,2,2,(2,2,3,1)i,2)	7 + 8i	5 + 8i	7 + 8i	-	-	-	*			-

UMS

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
UMS	11101101 11001000 {CY:DE':(HL)} = (IX) - [(IY) * DE + DE' + CY]; BC = BC-1; IX = IX+1; IY = IY+1; HL = HL+1; repeat while BC !=0	8+8i (2,2,2,(2,2,3,1)i,2)	7 + 8i	5 + 8i	7 + 8i	-	-	-	*			-

XOR

22 forms

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
XOR (HL)	10101110 A = [A & ~(HL)] [~A & (HL)]	5 (2,1,2)	5	5	5	*	*	P	0	fr	r	s
XOR (IX+d)	11011101 10101110 ----d--- A = [A & ~(IX+d)] [~A & (IX+d)]	10 (2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	s
XOR (IY+d)	11111101 10101110 ----d--- A = [A & ~(IY+d)] [~A & (IY+d)]	10 (2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	s
XOR n	11101110 ----n--- A = [A & ~n] [~A & n]	4 (2,2)	4	2	4	*	*	P	0	fr	r	
XOR r	10101-r- A = [A & ~r] [~A & r]	2	2	2	2	*	*	P	0	fr	r	
XOR HL,DE	01010100 HL = HL ^ DE	2	2	2	2	*	*	P	0	fr	r	
XOR (HL)	01111111 10101110 A = [A & ~(HL)] [~A & (HL)]	7 (2,2,1,2)	7	5	7	*	*	P	0	fr	r	s
XOR A	10101111 A = 0	2	2	2	2	*	*	P	0	fr	r	
XOR r	01111111 10101-r- A = [A & ~r] [~A & r]	4 (2,2)	4	2	4	*	*	P	0	fr	r	
XOR JKHL,BCDE	11101101 11101110 JKHL = JKHL ^ BCDE	4 (2,2)	4	2	4	*	*	P	0	fr	r	
XOR HL,DE	01111111 01010100 HL = HL ^ DE	4 (2,2)	4	2	4	*	*	P	0	fr	r	
XOR HL,(IX+d)	11011101 10100001 ----d--- HL = [HL & ~(IX+d)] [~HL & (IX+d)]	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	s
XOR JKHL,(IX+d)	11011101 10100011 ----d--- JKHL = [JKHL & ~(IX+d)] [~JKHL & (IX+d)]	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr	r	s
XOR HL,(IY+d)	11111101 10100001 ----d--- HL = [HL & ~(IY+d)] [~HL & (IY+d)]	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	s
XOR JKHL,(IY+d)	11111101 10100011 ----d--- JKHL = [JKHL & ~(IY+d)] [~JKHL & (IY+d)]	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr	r	s
XOR JKHL,ps	01001001 010100ps JKHL = [JKHL & ~ps] [~JKHL & ps]	4 (2,2)	4	2	4	*	*	P	0	fr	r	
XOR A,(ps+d)	01001001 010101ps ----d--- A = [A & ~(ps+d)] [~A & (ps+d)]	10 (2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	
XOR HL,(ps+d)	01001001 010110ps ----d--- HL = [HL & ~(ps+d)] [~HL & (ps+d)]	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	
XOR JKHL,(ps+d)	01001001 010111ps ----d--- JKHL = [JKHL & ~(ps+d)] [~JKHL & (ps+d)]	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr	r	
XOR A,(sp+n)	01001001 11011001 ----n--- A = [A & ~(sp+n)] [~A & (sp+n)]	10 (2,2,2,2,2)	10	9	8	*	*	P	0	fr	r	
XOR HL,(sp+n)	01001001 11011010 ----n--- HL = [HL & ~(sp+n)] [~HL & (sp+n)]	12 (2,2,2,2,2,2)	12	9	10	*	*	P	0	fr	r	

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
XOR JKHL, (sp+n)	01001001 11011011 ----n--- JKHL = [JKHL & ~(sp+n)] [~JKHL & (sp+n)]	16 (2,2,2,2,2,2,2,2)	16	11	14	*	*	P	0	fr		r

ZDMA

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
ZDMA	– dma transfer	?	?	?	?	-	-	-	-			

ZINTACK

1 form

Instruction	Encoding (binary, by byte)	Clocks				Flags				Prefixes		
		Breakdown	8-bit	16-bit align.	16-bit unal.	S	Z	PV	C	AD	AS	IO
ZINTACK	– (SP-1) = PCH; (SP-2) = PCL; SP = SP-2; IP = {IP[6:0], new IP}; PC = (vector table)	11 (2,3,3,3)	6	4	6	-	-	-	-			