

ZX SPECTRUM - Z80 Assembler

Mnemonic	Clock	Size	SZHPNC	OP-Code	Description	Notes
ADC A,r	4	1	***V0*	88+rb	Add with Carry	A=A+s+CY
ADC A,N	7	2		CE XX		
ADC A,(HL)	7	1		8E		
ADC A,(IX+N)	19	3		DD 8E XX		
ADC A,(IY+N)	19	3		FD 8E XX		
ADC HL,BC	15	2	**?V0*	ED 4A	Add with Carry	HL=HL+ss+CY
ADC HL,DE	15	2		ED 5A		
ADC HL,HL	15	2		ED 6A		
ADC HL,SP	15	2		ED 7A		
ADD A,r	4	1	***V0*	80+rb	Add (8-bit)	A=A+s
ADD A,N	7	2		C6 XX		
ADD A,(HL)	7	1		86		
ADD A,(IX+N)	19	3		DD 86 XX		
ADD A,(IY+N)	19	3		FD 86 XX		
ADD HL,BC	11	1	--?-0*	09	Add (16-bit)	HL=HL+ss
ADD HL,DE	11	1		19		
ADD HL,HL	11	1		29		
ADD HL,SP	11	1		39		
ADD IX,BC	15	2	--?-0*	DD 09	Add (IX register)	IX=IX+pp
ADD IX,DE	15	2		DD 19		
ADD IX,IX	15	2		DD 29		
ADD IX,SP	15	2		DD 39		
ADD IY,BC	15	2	--?-0*	FD 09	Add (IY register)	IY=IY+rr
ADD IY,DE	15	2		FD 19		
ADD IY,IX	15	2		FD 29		
ADD IY,SP	15	2		FD 39		
AND r	4	1	***P00	A0+rb	Logical AND	A=A&s
AND N	7	2		E6 XX		
AND (HL)	7	1		A6		
AND (IX+N)	19	3		DD A6 XX		
AND (IY+N)	19	3		FD A6 XX		
BIT b,r	8	2	?*1?0-	CB 40+8*b+rb	Test Bit	m&{2^b}
BIT b,(HL)	12	2		CB 46+8*b		
BIT b,(IX+N)	20	4		DD CB XX 46+8*b		
BIT b,(IY+N)	20	4		FD CB XX 46+8*b		
CALL NN	17	3	-----	CD XX XX	Unconditional Call	-(SP)=PC,PC=nn
CALL C,NN	17/1	3	-----	DC XX XX	Conditional Call	If Carry = 1
CALL NC,NN	17/1	3		D4 XX XX		If carry = 0
CALL M,NN	17/1	3		FC XX XX		If Sign = 1 (negative)
CALL P,NN	17/1	3		F4 XX XX		If Sign = 0 (positive)
CALL Z,NN	17/1	3		CC XX XX		If Zero = 1 (ans.=0)
CALL NZ,NN	17/1	3		C4 XX XX		If Zero = 0 (non-zero)
CALL PE,NN	17/1	3		EC XX XX		If Parity = 1 (even)
CALL PO,NN	17/1	3		E4 XX XX		If Parity = 0 (odd)
CCF	4	1	--?-0*	3F	Complement Carry Flag	CY=~CY
CP r	4	1	***V1*	B8+rb	Compare	Compare A-s
CP N	7	2		FE XX		
CP (HL)	7	1		BE		
CP (IX+N)	19	3		DD BE XX		
CP (IY+N)	19	3		FD BE XX		
CPD	16	2	****1-	ED A9	Compare and Decrement	A-(HL),HL=HL-1,BC=BC-1
CPDR	21/1	2	****1-	ED B9	Compare, Dec., Repeat	CPD till A=(HL)or BC=0
CPI	16	2	****1-	ED A1	Compare and Increment	A-(HL),HL=HL+1,BC=BC-1
CPIR	21/1	2	****1-	ED B1	Compare, Inc., Repeat	CPI till A=(HL)or BC=0
CPL	4	1	--1-1-	2F	Complement	A=~A
DAA	4	1	***P-*	27	Decimal Adjust Acc.	A=BCD format (dec.)

Mnemonic	Clock	Size	SZHPNC	OP-Code	Description	Notes
DEC A	4	1	***V1-	3D	Decrement (8-bit)	s=s-1
DEC B	4	1		05		
DEC C	4	1		0D		
DEC D	4	1		15		
DEC E	4	1		1D		
DEC H	4	1		25		
DEC L	4	2		2D		
DEC (HL)	11	1		35		
DEC (IX+N)	23	3		DD 35 XX		
DEC (IY+N)	23	3		FD 35 XX		
DEC BC	6	1	-----	0B	Decrement (16-bit)	ss=ss-1
DEC DE	6	1		1B		
DEC HL	6	1		2B		
DEC SP	6	1		3B		
DEC IX	10	2	-----	DD 2B	Decrement	xx=xx-1
DEC IY	10	2		FD 2B		
DI	4	1	-----	F3	Disable Interrupts	
DJNZ \$+2	13/8	1	-----	10	Dec., Jump Non-Zero	B=B-1 till B=0
EI	4	1	-----	FB	Enable Interrupts	
EX (SP),HL	19	1	-----	E3	Exchange	(SP)<->HL
EX (SP),IX	23	2	-----	DD E3		(SP)<->xx
EX (SP),IY	23	2		FD E3		
EX AF,AF'	4	1	-----	08		AF<->AF'
EX DE,HL	4	1	-----	EB		DE<->HL
EXX	4	1	-----	D9	Exchange	qq<->qq' (except AF)
HALT	4	1	-----	76	Halt	
IM 0	8	2	-----	ED 46	Interrupt Mode	(n=0,1,2)
IM 1	8	2		ED 56		
IM 2	8	2		ED 5E		
IN A,(N)	11	2	-----	DB XX	Input	A=(n)
IN (C)	12	2	***P0-	ED 70	Input*	(Unsupported)
IN A,(C)	12	2	***P0-	ED 78	Input	r=(C)
IN B,(C)	12	2		ED 40		
IN C,(C)	12	2		ED 48		
IN D,(C)	12	2		ED 50		
IN E,(C)	12	2		ED 58		
IN H,(C)	12	2		ED 60		
IN L,(C)	12	2		ED 68		
INC A	4	1	***V0-	3C	Increment (8-bit)	r=r+1
INC B	4	1		04		
INC C	4	1		0C		
INC D	4	1		14		
INC E	4	1		1C		
INC H	4	1		24		
INC L	4	1		2C		
INC BC	6	1	-----	03	Increment (16-bit)	ss=ss+1
INC DE	6	1		13		
INC HL	6	1		23		
INC SP	6	1		33		
INC IX	10	2	-----	DD 23	Increment	xx=xx+1
INC IY	10	2		FD 23		
INC (HL)	11	1	***V0-	34	Increment (indirect)	(HL)=(HL)+1
INC (IX+N)	23	3	***V0-	DD 34 XX	Increment	(xx+d)=(xx+d)+1
INC (IY+N)	23	3		FD 34 XX		
IND	16	2	?*??1-	ED AA	Input and Decrement	(HL)=(C),HL=HL-1,B=B-1
INDR	21/1	2	?1??1-	ED BA	Input, Dec., Repeat	IND till B=0
INI	16	2	?*??1-	ED A2	Input and Increment	(HL)=(C),HL=HL+1,B=B-1
INIR	21/1	2	?1??1-	ED B2	Input, Inc., Repeat	INI till B=0

Mnemonic	Clck	Siz	SZHPNC	OP-Code	Description	Notes
JP \$NN	10	3	-----	C3 XX XX	Unconditional Jump	PC=nn
JP (HL)	4	1	-----	E9	Unconditional Jump	PC=(HL)
JP (IX)	8	2	-----	DD E9	Unconditional Jump	PC=(xx)
JP (IY)	8	2		FD E9		
JP C,\$NN	10/1	3	-----	DA XX XX	Conditional Jump	If Carry = 1
JP NC,\$NN	10/1	3		D2 XX XX		If Carry = 0
JP M,\$NN	10/1	3		FA XX XX		If Sign = 1 (negative)
JP P,\$NN	10/1	3		F2 XX XX		If Sign = 0 (positive)
JP Z,\$NN	10/1	3		CA XX XX		If Zero = 1 (ans.= 0)
JP NZ,\$NN	10/1	3		C2 XX XX		If Zero = 0 (non-zero)
JP PE,\$NN	10/1	3		EA XX XX		If Parity = 1 (even)
JP PO,\$NN	10/1	3		E2 XX XX		If Parity = 0 (odd)
JR \$N+2	12	2	-----	18 XX	Relative Jump	PC=PC+e
JR C,\$N+2	12/7	2	-----	38 XX	Cond. Relative Jump	If cc JR(cc=C,NC,NZ,Z)
JR NC,\$N+2	12/7	2		30 XX		
JR Z,\$N+2	12/7	2		28 XX		
JR NZ,\$N+2	12/7	2		20 XX		
LD I,A	9	2	-----	ED 47	Load*	dst=src
LD R,A	9	2		ED 4F		
LD A,I	9	2	**0*0-	ED 57	Load*	dst=src
LD A,R	9	2		ED 5F		
LD A,r	4	1	-----	78+rb	Load (8-bit)	dst=src
LD A,N	7	2		3E XX		
LD A,(BC)	7	1		0A		
LD A,(DE)	7	1		1A		
LD A,(HL)	7	1		7E		
LD A,(IX+N)	19	3		DD 7E XX		
LD A,(IY+N)	19	3		FD 7E XX		
LD A,(NN)	13	3		3A XX XX		
LD B,r	4	1		40+rb		
LD B,N	7	2		06 XX		
LD B,(HL)	7	1		46		
LD B,(IX+N)	19	3		DD 46 XX		
LD B,(IY+N)	19	3		FD 46 XX		
LD C,r	4	1		48+rb		
LD C,N	7	2		0E XX		
LD C,(HL)	7	1		4E		
LD C,(IX+N)	19	3		DD 4E XX		
LD C,(IY+N)	19	3		FD 4E XX		
LD D,r	4	1		50+rb		
LD D,N	7	2		16 XX		
LD D,(HL)	7	1		56		
LD D,(IX+N)	19	3		DD 56 XX		
LD D,(IY+N)	19	3		FD 56 XX		
LD E,r	4	1		58+rb		
LD E,N	7	2		1E XX		
LD E,(HL)	7	1		5E		
LD E,(IX+N)	19	3		DD 5E XX		
LD E,(IY+N)	19	3		FD 5E XX		
LD H,r	4	1		60+rb		
LD H,N	7	2		26 XX		
LD H,(HL)	7	1		66		
LD H,(IX+N)	19	3		DD 66 XX		
LD H,(IY+N)	19	3		FD 66 XX		
LD L,r	4	1		68+rb		
LD L,N	7	2		2E XX		
LD L,(HL)	7	1		6E		
LD L,(IX+N)	19	3		DD 6E XX		
LD L,(IY+N)	19	3		FD 6E XX		

(LD continues in next page)

Mnemonic	Clock	Size	SZHPNC	OP-Code	Description	Notes
LD BC, (NN)	20	4	-----	ED 4B XX XX	Load (16-bit)	dst=src
LD BC, NN	10	3		01 XX XX		
LD DE, (NN)	20	4		ED 5B XX XX		
LD DE, NN	10	3		11 XX XX		
LD HL, (NN)	20	3		2A XX XX		
LD HL, NN	10	3		21 XX XX		
LD SP, (NN)	20	4		ED 7B XX XX		
LD SP, HL	6	1		F9		
LD SP, IX	10	2		DD F9		
LD SP, IY	10	2		FD F9		
LD SP, NN	10	3		31 XX XX		
LD IX, (NN)	20	4		DD 2A XX XX		
LD IX, NN	14	4		DD 21 XX XX		
LD IY, (NN)	20	4		FD 2A XX XX		
LD IY, NN	14	4		FD 21 XX XX		
LD (HL), r	7	1	-----	70+rb	Load (Indirect)	dst=src
LD (HL), N	10	2		36 XX		
LD (BC), A	7	1		02		
LD (DE), A	7	1		12		
LD (NN), A	13	3		32 XX XX		
LD (NN), BC	20	4		ED 43 XX XX		
LD (NN), DE	20	4		ED 53 XX XX		
LD (NN), HL	16	3		22 XX XX		
LD (NN), IX	20	4		DD 22 XX XX		
LD (NN), IY	20	4		FD 22 XX XX		
LD (NN), SP	20	4		ED 73 XX XX		
LD (IX+N), r	19	3		DD 70+rb XX		
LD (IX+N), N	19	4		DD 36 XX XX		
LD (IY+N), r	19	3		FD 70+rb XX		
LD (IY+N), N	19	4		FD 36 XX XX		
LDD	16	2	--0*0-	ED A8	Load and Decrement	(DE)=(HL), HL=HL-1, #
LDDR	21/1	2	--000-	ED B8	Load, Dec., Repeat	LDD till BC=0
LDI	16	2	--0*0-	ED A0	Load and Increment	(DE)=(HL), HL=HL+1, #
LDIR	21/1	2	--000-	ED B0	Load, Inc., Repeat	LDI till BC=0
NEG	8	2	***V1*	ED 44	Negate	A=-A
NOP	4	1	-----	00	No Operation	
OR r	4	1	***P00	B0+rb	Logical inclusive OR	A=Avs
OR N	7	2		F6 XX		
OR (HL)	7	1		B6		
OR (IX+N)	19	3		DD B6 XX		
OR (IY+N)	19	3		FD B6 XX		
OUT (N), A	11	2	-----	D3 XX	Output	(n)=A
OUT (C), 0	12	2	-----	ED 71	Output*	(Unsupported)
OUT (C), A	12	2	-----	ED 79	Output	(C)=r
OUT (C), B	12	2		ED 41		
OUT (C), C	12	2		ED 49		
OUT (C), D	12	2		ED 51		
OUT (C), E	12	2		ED 59		
OUT (C), H	12	2		ED 61		
OUT (C), L	12	2		ED 69		
OUTD	16	2	?*??1-	ED AB	Output and Decrement	(C)=(HL), HL=HL-1, B=B-1
OTDR	21/1	2	?1??1-	ED BB	Output, Dec., Repeat	OUTD till B=0
OUTI	16	2	?*??1-	ED A3	Output and Increment	(C)=(HL), HL=HL+1, B=B-1
OTIR	21/1	2	?1??1-	ED B3	Output, Inc., Repeat	OUTI till B=0
POP AF	10	1	-----	F1	Pop	qq=(SP)+
POP BC	10	1		C1		
POP DE	10	1		D1		
POP HL	10	1		E1		
POP IX	14	2	-----	DD E1	Pop	xx=(SP)+
POP IY	14	2		FD E1		
PUSH AF	11	1	-----	F5	Push	-(SP)=qq
PUSH BC	11	1		C5		
PUSH DE	11	1		D5		
PUSH HL	11	1		E5		
PUSH IX	15	2	-----	DD E5	Push	-(SP)=xx
PUSH IY	15	2		FD E5		

Mnemonic	Clock	Size	SZHPNC	OP-Code	Description	Notes
RES b,r	8	2	-----	CB 80+8*b+rb	Reset bit	m=m&{~2^b}
RES b,(HL)	15	2	-----	CB 86+8*b		
RES b,(IX+N)	23	4	-----	DD CB XX 86+8*b		
RES b,(IY+N)	23	4	-----	FD CB XX 86+8*b		
RET	10	1	-----	C9	Return	PC=(SP)+
RET C	11/5	1	-----	D8	Conditional Return	If Carry = 1
RET NC	11/5	1		D0		If Carry = 0
RET M	11/5	1		F8		If Sign = 1 (negative)
RET P	11/5	1		F0		If Sign = 0 (positive)
RET Z	11/5	1		C8		If Zero = 1 (ans.=0)
RET NZ	11/5	1		C0		If Zero = 0 (non-zero)
RET PE	11/5	1		E8		If Parity = 1 (even)
RET PO	11/5	1		E0		If Parity = 0 (odd)
RETI	14	2	-----	ED 4D	Return from Interrupt	PC=(SP)+
RETN	14	2	-----	ED 45	Return from NMI	PC=(SP)+
RLA	4	1	--0-0*	17	Rotate Left Acc.	A={CY,A}<-
RL r	8	2	**0P0*	CB 10+rb	Rotate Left	m={CY,m}<-
RL (HL)	15	2		CB 16		
RL (IX+N)	23	4		DD CB XX 16		
RL (IY+N)	23	4		FD CB XX 16		
RLCA	4	1	--0-0*	07	Rotate Left Cir. Acc.	A=A<-
RLC r	8	2	**0P0*	CB 00+rb	Rotate Left Circular	m=m<-
RLC (HL)	15	2		CB 06		
RLC (IX+N)	23	4		DD CB XX 06		
RLC (IY+N)	23	4		FD CB XX 06		
RLD	18	2	**0P0-	ED 6F	Rotate Left 4 bits	{A,(HL)}={A,(HL)}<- ##
RRA	4	1	--0-0*	1F	Rotate Right Acc.	A-->{CY,A}
RR r	8	2	**0P0*	CB 18+rb	Rotate Right	m-->{CY,m}
RR (HL)	15	2		CB 1E		
RR (IX+N)	23	4		DD CB XX 1E		
RR (IY+N)	23	4		FD CB XX 1E		
RRCA	4	1	--0-0*	0F	Rotate Right Cir.Acc.	A-->A
RRC r	8	2	**0P0*	CB 08+rb	Rotate Right Circular	m-->m
RRC (HL)	15	2		CB 0E		
RRC (IX+N)	23	4		DD CB XX 0E		
RRC (IY+N)	23	4		FD CB XX 0E		
RRD	18	2	**0P0-	ED 67	Rotate Right 4 bits	{A,(HL)}=->{A,(HL)} ##
RST 0	11	1	-----	C7	Restart	(p=0H,8H,10H,...,38H)
RST 08H	11	1		CF		
RST 10H	11	1		D7		
RST 18H	11	1		DF		
RST 20H	11	1		E7		
RST 28H	11	1		EF		
RST 30H	11	1		F7		
RST 38H	11	1		FF		
SBC r	4	1	***V1*	98+rb	Subtract with Carry	A=A-s-CY
SBC A,N	7	2		DE XX		
SBC (HL)	7	1		9E		
SBC A,(IX+N)	19	3		DD 9E XX		
SBC A,(IY+N)	19	3		FD 9E XX		
SBC HL,BC	15	2	**?V1*	ED 42	Subtract with Carry	HL=HL-ss-CY
SBC HL,DE	15	2		ED 52		
SBC HL,HL	15	2		ED 62		
SBC HL,SP	15	2		ED 72		
SCF	4	1	--0-01	37	Set Carry Flag	CY=1
SET b,r	8	2	-----	CB C0+8*b+rb	Set bit	m=mv{2^b}
SET b,(HL)	15	2		CB C6+8*b		
SET b,(IX+N)	23	4		DD CB XX C6+8*b		
SET b,(IY+N)	23	4		FD CB XX C6+8*b		
SLA r	8	2	**0P0*	CB 20+rb	Shift Left Arithmetic	m=m*2
SLA (HL)	15	2		CB 26		
SLA (IX+N)	23	4		DD CB XX 26		
SLA (IY+N)	23	4		FD CB XX 26		
SRA r	8	2	**0P0*	CB 28+rb	Shift Right Arith.	m=m/2
SRA (HL)	15	2		CB 2E		
SRA (IX+N)	23	4		DD CB XX 2E		
SRA (IY+N)	23	4		FD CB XX 2E		

Mnemonic	Clck	Siz	SZHPNC	OP-Code	Description	Notes
SLL r	8	2	**0P0*	CB 30+rb	Shift Left Logical*	m={0,m,CY}<-
SLL (HL)	15	2		CB 36		(SLL instructions are Unsupported)
SLL (IX+N)	23	4		DD CB XX 36		
SLL (IY+N)	23	4		FD CB XX 36		
SRL r	8	2	**0P0*	CB 38+rb	Shift Right Logical	m=->{0,m,CY}
SRL (HL)	15	2		CB 3E		
SRL (IX+N)	23	4		DD CB XX 3E		
SRL (IY+N)	23	4		FD CB XX 3E		
SUB r	4	1	***V1*	90+rb	Subtract	A=A-s
SUB N	7	2		D6 XX		
SUB (HL)	7	1		96		
SUB (IX+N)	19	3		DD 96 XX		
SUB (IY+N)	19	3		FD 96 XX		
XOR r	4	1	***P00	A8+rb	Logical Exclusive OR	A=Axs
XOR N	7	2		EE XX		
XOR (HL)	7	1		AE		
XOR (IX+N)	19	3		DD AE XX		
XOR (IY+N)	19	3		FD AE XX		

n	Immediate addressing
nn	Immediate extended addressing
e	Relative addressing (PC=PC+2+offset)
(nn)	Extended addressing
(xx+d)	Indexed addressing
r	Register addressing
(rr)	Register indirect addressing
	Implied addressing
b	Bit addressing
p	Modified page zero addressing (see RST)
*	Undocumented opcode
A B C D E	Registers (8-bit)
AF BC DE HL	Register pairs (16-bit)
F	Flag register (8-bit)
I	Interrupt page address register (8-bit)
IX IY	Index registers (16-bit)
PC	Program Counter register (16-bit)
R	Memory Refresh register
SP	Stack Pointer register (16-bit)
b	One bit (0 to 7)
cc	Condition (C,M,NC,NZ,P,PE,PO,Z)
d	One-byte expression (-128 to +127)
dst	Destination s, ss, (BC), (DE), (HL), (nn)
e	One-byte expression (-126 to +129)
m	Any register r, (HL) or (xx+d)
n	One-byte expression (0 to 255)
nn	Two-byte expression (0 to 65535)
pp	Register pair BC, DE, IX or SP
qq	Register pair AF, BC, DE or HL
qq'	Alternative register pair AF, BC, DE or HL
r	Register A, B, C, D, E, H or L
rr	Register pair BC, DE, IY or SP
s	Any register r, value n, (HL) or (xx+d)
src	Source s, ss, (BC), (DE), (HL), nn, (nn)
ss	Register pair BC, DE, HL or SP
xx	Index register IX or IY
+ - * / ^	Add/subtract/multiply/divide/exponent
& ~ v x	Logical AND/NOT/inclusive OR/exclusive OR
<- ->	Rotate left/right
()	Indirect addressing
()+ -()	Indirect addressing auto-increment/decrement
{ }	Combination of operands
#	Also BC=BC-1,DE=DE-1
##	Only lower 4 bits of accumulator A used