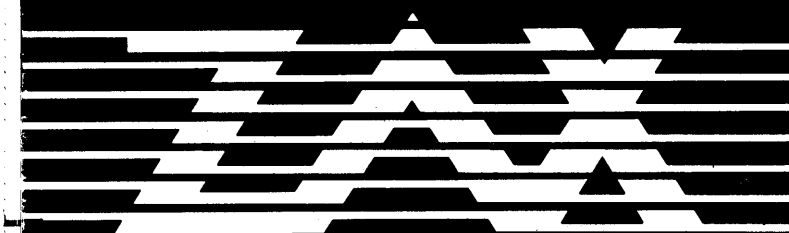
Command Reference Guide

ICD-278 FOR Z80



ZAX

Zax Corporation 2572 White Road, Irvine, California 92714
Toll Free 1-800-421-0982 CA, HA, AK 714-474-1170

INTRODUCTION

This COMMAND REFERENCE GUIDE shows the command formats that are available with the ICD-278 for Z80.

This COMMAND REFERENCE GUIDE is intended for the experienced ICD operator and contains the command parameters only. If you are unfamiliar with the parameter context, refer to the MASTER COMMAND GUIDE in the USER'S MANUAL for a detailed explanation of each command.

General reference information about the Z80 processor is located on the back of this GUIDE.

NOTES ON COMMAND FORMATS

Items which are designated as KEYWORDS are displayed by BOLD characters (such as M, ON, OFF). They represent items which you must enter. Any combination of upper/lower case letters may be used.

Lowercase letters show items which you must supply (USER-SUPPLIED ITEMS), such as "filename" in which you would enter the name of a file you've selected.

A vertical line ("|") between KEYWORDS (ON | OFF, ARM | IND) means you must select one of the items. For example, if the command contains HI | LO, HI or LO may be entered but not both.

Items shown in brackets ("[ ]") indicate that the parameter is OPTIONAL.

Include all punctuation such as commas (,), equal signs (=), colons (:), and slashes (/).

Some parameters are abbreviated as follows:

ipt/tpt = initiation/termination pointer
ua = user address
pc = passcount

EMULATION MODE

In-circuit (I)	Sets or displays in-circuit mode.
Map (MA)	Sets or displays memory map conditions.
Pin (PI)	Sets or displays CPU pin status.

DEBUGGING & BREAK/EVENT

Break (B)	Sets or displays breakpoint conditions.
Event (EV)	Sets or displays event point conditions.
Go (G)	Begins program execution.
Next (N)	Traces program in "n" steps.
Trace (T)	Sets or displays the trace mode conditions.

MEMORY:I/O:REGISTER

Assemble (A)	Writes assembly language to memory.
Compare (CO)	Compares data between user and program memory.
Disassemble (DI)	Displays memory contents in mnemonic code.
Dump (D)	Displays hexadecimal memory contents.
Examine (E)	Displays memory contents in either ASCII code or hex data.
Fill (F)	Fills memory with ASCII code or hex data.
Move (M)	Moves data between the target memory and program memory.
Port (P)	Examines and/or changes I/O port contents.
Register (R)	Changes or displays the processor registers.
Search (S)	Searches memory contents.

PROGRAM INPUT/OUTPUT

Load (L)	Loads object program from host system through a specified port.
Save (SA)	Saves memory contents to a disk file on the host system.
Verify (V)	Compares an Intel format file to the ICD memory contents.

REAL-TIME TRACE

History (H)	Displays the contents of the real-time trace buffer.

OTHERS

Identify (ID)	Displays current ICD device name and firmware version.
Offset (O)	Sets value in offset register for relative addressing.
Print (PR)	Controls output of ICD commands to an external printer.
* Quit (Q)	Reboots to host system.
Supervisor (SU)	Sets a breakpoint as a supervisor call.
User (U)	Allows terminal console to be used for host computer console.
* Z (Z)	Expands operation commands.

TECHNICAL PUBLICATION NO. 20-602-01

* AVAILABLE WITH ZICE SOFTWARE ONLY

ASSEMBLE specification:
>A mem_addr (cr)
xxxx (Z80 assembler code) (cr)
xxxx (cr)

□ >A 100
0100 LD A,(HL)
0101

BREAK status:
>B

Hardware BREAK specification:
>B[/A|/B|/C] M|MR| PW| OF,addr[,pc]
P | MW|PR |IA

□ >B/C MR,100,1

Hardware BREAK enable/disable:
>B[/A] [/B] [/C] ON|OFF|CLR

□ >B/A ON

Hardware arm/individual BREAK specification:
>B[/A] [/B] [/C] ARM|IND

□ >B/A ARM

BREAK initialize—clear Event Done flag:
>B INI

Software BREAK specification:
>B[/0] [/1] [/2]. . . [/7] addr[,pc]

□ >B/7 100,1

Software BREAK enable/disable:
>B[/0] [/1] [/2]. . . [/7] ON|OFF|CLR

□ >B/3 ON

Software BREAK op code or enable/disable:
>B S=op_code | EN | DI

□ >B S=EN

External BREAK HI/LOW:
>B/X HI|LO

□ >B/X LO

External BREAK enable/disable:
>B/X ON|OFF|CLR

□ >B/X CLR

Event BREAK enable/disable:
>B/E ON|OFF

□ >B/E OFF

Timeout BREAK enable/disable:
>B/T ON|OFF

□ >B/T ON

Write protect BREAK enable/disable:
>B/W ON|OFF

□ >B/W OFF

COMPARE:
>CO beg_addr,end_addr,comp_addr[,UP|,PU]

□ >CO 100,3FF,20,UP

DISASSEMBLE:
>DI [beg_addr] [,end_addr]

□ >DI 100,200

DUMP:
>D[/W] beg_addr [,end addr]

□ >D/W 100,200

EVENT status:
>EV

EVENT specification:
>EV [ST=M|MR |MW|PR] [,A=addr] [,D=data]
[ST=P | PW|OP |IA]
[ST=ANY]

□ >EV ST=MR,A=100,D=55

EVENT enable/disable:
>EV ON|OFF|CLR

□ >EV CLR

EXAMINE memory only:
>E[/W] [/N] addr

□ >E/W 100

EXAMINE and change:
>E[/W] beg_addr=data data data . . .

□ >E/W 100=5555 8888

FILL:
>F[/W] [/N] beg_addr,end_addr,data data data . . .

□ >F/W 100,3FF,5555

GO:
>G beg_addr [,end_addr] [,end_addr#2]

□ >G 100

Realtime trace status (HISTORY):
>H

Realtime trace counter reset (HISTORY):
>H CLR

Realtime trace buffer storage mode (HISTORY):
>H BM|EM|BE|CE|EE|ME[,range]

□ >H ME,500

HISTORY search:
>H S,/[addr]/[data]/[IA | MR] [,ipt] [,tpt]
[MW | PR]
[M1]

□ >H S,/100/55/IA,200,100

□ SHOWS COMMAND EXAMPLE

HISTORY format display:
>H M|D[,ipt] [,tpt]
☐ >H M,200,100

IDENTIFICATION:
>ID

IN-CIRCUIT status:
>I

IN-CIRCUIT specification:
>I 0|1|2 (cr)
☐ >I 2

***LOAD:**
>L[/T|/P|/A|/H] filename[.ftype] [,bias]
☐ >L/T TEST.HEX,100

MAPPING status:
>MA

MAPPING specification:
>MA beg_addr [,end_addr] =RO|RW|NO|US
☐ >MA 100,200=RW

MOVE:
>M beg_addr,end_addr,mov_addr[,UP|,PU]
☐ >M 100,3FF,1000,UP

NEXT:
>N steps
☐ >N 5

OFFSET status:
>O

OFFSET specification:
>O &1|&2|&3|&4=addr
☐ >O &2=100

PIN status:
>PI

PIN specification:
>PI BUSRQ|NMI|INTR=EN|DI
☐ >PI BUSRQ=EN

PORT examination:
>P beg_addr
☐ >P 55

PORT examination and change:
>P port_addr=data data data . . .
☐ >P FF=12 12 34 56 78

PRINT:
>PR ON | OFF
☐ >PR ON

***QUIT:**
>Q

REGISTER:
>R

REGISTER reset:
>R RESET

REGISTER change:
>R reg types=new value
reg types include A A' PC SP IX IY
B C BC B' C' BC'
D E DE D' E' DE'
H L HL H' L' HL'
I IF F F' S Z P N CY
☐ >R HL=A000

***SAVE:**
>SA[/T|/P|/A|/H] filename[.ftype],beg addr,end_addr[,ua]
☐ >SA/T TEST.HEX,0,3FF,0

SEARCH:
>S[/W] [/D] beg addr,end addr,data data data . . .
☐ >S/W 100,200,5555

SUPERVISOR status:
>SU

SUPERVISOR specification:
>SU[/C|/7|/U] ON|OFF
☐ >SU/7 ON

TRACE status:
>T

TRACE specification:
>T[/S] A|J [,beg_addr] [,end_addr]
☐ >T/S J,100 300

TRACE enable/disable:
>T ON|OFF|CLR
☐ >T ON

USER:
>U code
☐ > U !

***VERIFY:**
>V[/T|/P|/A|/H] filename[.ftype] [,bias]
☐ >V/T TEST.HEX,100

***ZICE:**
>Z B|C|D|H|M|P|R|W

INSTRUCTION SET

8-BIT LOAD GROUP

Mnemonic	Symbolic Operation
LD r, s	r ← s
LD r, n	r ← n
LD r, (HL)	r ← (HL)
LD r, (IX + d)	r ← (IX + d)
LD r, (IY + d)	r ← (IY + d)
LD (HL), r	(HL) ← r
LD (IX + d), r	(IX + d) ← r
LD (IY + d), r	(IY + d) ← r
LD (HL), n	(HL) ← n
LD (IX + d), n	(IX + d) ← n
LD (IY + d), n	(IY + d) ← n
LD A, (BC)	A ← (BC)
LD A, (DE)	A ← (DE)
LD A, (nn)	A ← (nn)
LD (BC), A	(BC) ← A
LD (DE), A	(DE) ← A
LD (nn), A	(nn) ← A
LD A, I	A ← I
LD A, R	A ← R
LD I, A	I ← A
LD R, A	R ← A

S = Sign flag
Z = Zero flag
P/V = Parity/overflow flag
H = Half carry flag
N = Add/sub flag
C = Carry/link flag

16-BIT LOAD GROUP

Mnemonic	Symbolic Operation
LD dd, nn	dd ← nn
LD IX, nn	IX ← nn
LD IY, nn	IY ← nn
LD HL, (nn)	H ← (nn + 1)
	L ← (nn)
LD dd, (nn)	dd_H ← (nn + 1)
	dd_L ← (nn)
LD IX, (nn)	IX_H ← (nn + 1)
	IX_L ← (nn)
LD IY, (nn)	IY_H ← (nn + 1)
	IY_L ← (nn)
LD (nn), HL	(nn + 1) ← H
	(nn) ← L
LD (nn), dd	(nn + 1) ← dd_H
	(nn) ← dd_L
LD (nn), IX	(nn + 1) ← IX_H
	(nn) ← IX_L
(LD (nn), IY	(nn + 1) ← IY_H
	(nn) ← IY_L
LD SP, HL	SP ← HL
LD SP, IX	SP ← IX
LD SP, IY	SP ← IY
PUSH qq	(SP – 2) ← qq_L
	(SP – 1) ← qq_H
PUSH IX	(SP – 2) ← IX_L
	(SP – 1) ← IX_H
PUSH IY	(SP – 2) ← IY_L
	(SP – 1) ← IY_H
POP qq	qq_H ← (SP + 1)
	qq_L ← (SP)
POP IX	IX_H ← (SP + 1)
	IX_L ← (SP)
POP IY	IY_H ← (SP + 1)
	IY_L ← (SP)

INSTRUCTION SET (Cont.)

EXCHANGE GROUP AND BLOCK TRANSFER AND SEARCH GROUP

Mnemonic	Symbolic Operation	Mnemonic	Symbolic Operation
EX DE, HL	DE ↔ HL	LDDR	(DE) ← (HL)
EX AF, AF′	AF ↔ AF′		DE ← DE – 1
EXX	BC ↔ BC′		HL ← HL – 1
	DE ↔ DE′		BC ← BC – 1
	HL ↔ HL′		Repeat until
EX (SP), HL	H ↔ (SP + 1)		BC = 0
	L ↔ (SP)	CPI	A – (HL)
EX (SP), IX	IX_H ↔ (SP + 1)		HL ← HL + 1
	IX_L ↔ (SP)		BC ← BC – 1
EX (SP), IY	IY_H ↔ (SP + 1)	CPIR	A – (HL)
	IY_L ↔ (SP)		HL ← HL + 1
LDI	(DE) ← (HL)		BC ← BC – 1
	DE ← DE + 1		Repeat until
	HL ← HL + 1		A = (HL) or
	BC ← BC – 1		BC = 0
LDIR	(DE) ← (HL)	CPD	A – (HL)
	DE ← DE + 1		HL ← HL – 1
	HL ← HL + 1		BC ← BC – 1
	BC ← BC – 1	CPDR	A – (HL)
	Repeat until		HL ← HL – 1
	BC = 0		BC ← BC – 1
LDD	(DE) ← (HL)		Repeat until
	DE ← DE – 1		A = (HL) or
	HL ← HL – 1		BC = 0
	BC ← BC – 1		

n = (0–255) range
r = registers
s = location of Address
ii = index register
R = refresh
nn = (0–65535)
ss = 16 bit location, BC, DE, HL, SP
dd = pairs BC, DE, HL, SP
qq = pairs AF, BC, DE, HL
pp = pairs BC, DE, IX, SP
rr = pairs BC, DE, IY, SP
e = The extension in relative addressing mode

↕ flag affected by operation
• flag unchanged
O flag reset
L flag set
X don't care
V overflow
P parity

INSTRUCTION SET (Cont.)

8-BIT ARITHMETIC AND LOGICAL GROUP

Mnemonic	Symbolic Operation
ADD A, r	A ← A + r
ADD A, n	A ← A + n
ADD A, (HL)	A ← A + (HL)
ADD A, (IX + d)	A ← A + (IX + d)
ADD A, (IY + d)	A ← A + (IY + d)
ADC A, s	A ← A + s + CY
SUB s	A ← A − s
SBC A, s	A ← A − s − CY
AND s	A ← A △ s
OR s	A ← A ▽ s
XOR s	A ← A ⊕ s
CP s	A − s
INC r	r ← r + 1
INC (HL)	(HL) ← (HL) + 1
INC (IX + d)	(IX + d) ← (IX + d) + 1
INC (IY + d)	(IY + d) ← (IY + d) + 1
DEC s	s ← s − 1

16-BIT ARITHMETIC GROUP

Mnemonic	Symbolic Operation
ADD HL, ss	HL ← HL + ss
ADC HL, ss	HL ← HL + ss ↔ CY
SBC HL, ss	HL ← HL − ss − CY
ADD IX, pp	IX ← IX + pp
ADD IY, rr	IY ← IY + rr
INC ss	ss ← ss + 1
INC IX	IX ← IX + 1
INC IY	IY ← IY + 1
DEC ss	ss ← ss − 1
DEC IX	IX ← IX − 1
DEC IY	IY ← IY − 1

JUMP GROUP

Mnemonic	Symbolic Operation
JP nn	PC ← nn
JP cc, nn	If condition cc is true PC ← nn, otherwise continue
JR e	PC ← PC + e
JR C, e	If C = 0, continue If C = 1, PC ← PC + e
JR NC, e	If C = 1, continue If C = 0, PC ← PC + e
JR Z, e	If Z = 0 continue If Z = 1, PC ← PC + e
JR NZ, e	If Z = 1, continue If Z = 0, PC ← PC + e
JP (HL)	PC ← HL
JP (IX)	PC ← IX
JP (IY)	PC ← IY
DJNZ, e	B ← B − 1 If B = 0, continue If B ≠ 0, PC ← PC + e

BIT SET, RESET AND TEST GROUP

Mnemonic	Symbolic Operation
BIT b, r	$Z \leftarrow \overline{r_b}$
BIT b, (HL)	$Z \leftarrow \overline{(IX+d)_b}$
BIT b, (IX + d)$_b$	$Z \leftarrow \overline{(IX+d)_b}$
BIT b, (IY + d)$_b$	$Z \leftarrow \overline{(IY+d)_b}$
SET b, r	$r_b \leftarrow 1$
SET b, (HL)	$(HL)_b \leftarrow 1$
SET b, (IX + d)	$(IX+d)_b \leftarrow 1$
RES b, s	$s_b \leftarrow 0$ s ≡ r, (HL), (IX + d), (IY + d)

INSTRUCTION SET (Cont.)

CALL AND RETURN GROUP

Mnemonic	Symbolic Operation
CALL nn	(SP − 1) ← PC_H (SP − 2) ← PC_L PC ← nn
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn
RET	PC_L ← (SP) PC_H ← (SP + 1)
RET cc	If condition cc is false continue, otherwise same as RET
RETI	Return from interrupt
RETN[1]	Return from non maskable interrupt
RST p	(SP − 1) ← PC_H (SP − 2) ← PC_L PC_H ← 0 PC_L ← p

Mnemonic	Symbolic Operation
IN A, (n)	A ← (n)
IN r, (C)	r ← (C) if r = 110 only the flags will be affected
INI	(HL) ← (C) B ← B − 1 HL ← HL + 1
INIR	(HL) ← (C) B ← B − 1 HL ← HL + 1 Repeat until B = 0
IND	(HL) ← (C) B ← B − 1 HL ← HL − 1
INDR	(HL) ← (C) B ← B − 1 HL ← HL − 1 Repeat until B = 0
OUT (n), A	(n) ← A
OUT (C), r	(C) ← r
OUTI	(C) ← (HL) B ← B − 1 HL ← HL + 1
OTIR	(C) ← (HL) B ← B − 1 HL ← HL + 1 Repeat until B = 0
OUTD	(C) ← (HL) B ← B − 1 HL ← HL − 1
OTDR	(C) ← (HL) B ← B − 1 HL ← HL − 1 Repeat until B = 0

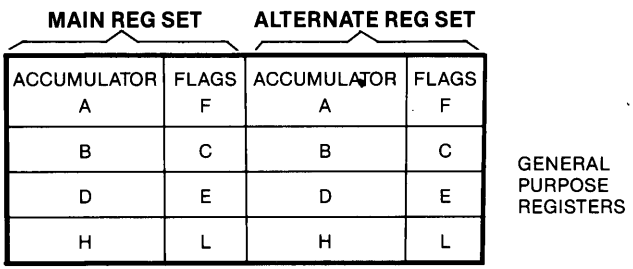
ASCII CHARACTER SET (7-Bit Code)

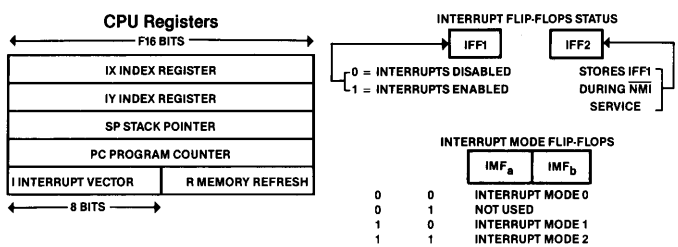
LSD \ MSD	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 0000	NUL	DLE	SP	0	@	P	‘	p
1 0001	SOH	DC1	!	1	A	Q	a	q
2 0010	STX	DC2	“	2	B	R	b	r
3 0011	ETX	DC3	#	3	C	S	c	s
4 0100	EOT	DC4	$	4	D	T	d	t
5 0101	ENQ	NAK	%	5	E	U	e	u
6 0110	ACK	SYN	&	6	F	V	f	v
7 0111	BEL	ETB	’	7	G	W	g	w
8 1000	BS	CAN	(	8	H	X	h	x
9 1001	HT	EM	)	9	I	Y	i	y
A 1010	LF	SUB	°	:	J	Z	j	z
B 1011	VT	ESC	+	;	K	[	k	{
C 1100	FF	FS	’	<	L	/	l	|
D 1101	CR	GS	-	=	M	]	m	}
E 1110	SO	RS	·	>	N	↑	n	~
F 1111	SI	US	/	?	O	←	o	DEL

Hex-Decimal Conversion

6		5		4		3		2		1	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
0123		4567		0123		4567		0123		4567	
BYTE				BYTE				BYTE			

Z80-CPU REGISTER CONFIGURATION





ERROR MESSAGES

ERROR MESSAGE	DISPLAYED WHEN;
UNABLE BREAK ADDRESS	a software break is specified in the non-RAM area
MULTI BREAK ADDRESS	a software break is duplicated at the same address
WARNING UNABLE SOFT BREAK	a software break is set at the address presently not mapped in RAM
***FILE NOT FOUND	no file exists
***DISK READ ERROR	a disk read error occurs
***DISK WRITE ERROR	a sum check error occurs
***NO DIRECTORY SPACE	no blank area available
C?>	a command code error occurs
P?>	a parameter code error occurs
/?>	a modifier code error occurs
MEMORY WRITE ERROR AT XXX	there is a memory modification error
MEMORY TIMEOUT ERROR AT XXXX	memory I/O in the target system does not respond to ICD access
I/O TIMEOUT ERROR AT XX	timeouts a waitstate
XXXX INPUT ERROR	an input error occurs
BREAK BUSY	the break specification exceeds the limit