

 OSBORNE/McGraw-Hill

# Z80

ASSEMBLY LANGUAGE PROGRAMMING



Lance A. Leventhal

*Scanned and converted to PDF by HansO, 2001*

Chapter 3 page 164-169

**8080A/Z80 compatability**

## 8080A/Z80 COMPATIBILITY

Although the Z80 microprocessor can certainly be used on its own merits, one of its important characteristics is its compatibility with the 8080A microprocessor. This compatibility has the following features:

**8080A/Z80  
COMPATIBILITY  
FEATURES**

- 1) All 8080A machine language instructions are also Z80 machine language instructions.
- 2) All 8080A registers are also Z80 registers (see Table 3-6).
- 3) Almost all 8080A programs will run on a Z80, with some minor differences to be noted later.
- 4) The Z80 has instructions, registers, and other features not present on the 8080A, so Z80 programs will not generally run on 8080A processors.

**Note that this compatibility does not extend to assembly language source statements** since Z80 assemblers and 8080A assemblers use different operation code mnemonics. **Table 3-7 contains a list of the 8080A mnemonic codes and the corresponding Z80 codes, while Table 3-8 is the same list organized by Z80 codes.**

**8080A/Z80  
ASSEMBLY  
LEVEL  
CONVERSION**

Readers should note the binary coding limitations that this compatibility places on the extra features of the Z80 microprocessor. The 8080A has some unused operation codes (see Table 3-9) that are used for some of the Z80's extra instructions. But there are simply not enough such codes to cover the large number of features in a simple form.

**8080A  
UNUSED  
OPERATION  
CODES**

Thus, many of the added Z80 instructions require a 2-byte operation code. The first byte is CB, DD, ED, or FD. Note the following meanings of these codes from Table 3-9:

**2-BYTE  
OPERATION  
CODES**

- CB — a register or bit operation
- DD — an operation involving register IX
- ED — a miscellaneous non-8080A instruction not covered elsewhere
- FD — an operation involving register IY

The second byte of the operation code describes the actual operation to be performed.

The end result is that these multi-byte instructions execute rather slowly (and use more memory) because an additional memory access is required. The reader should be aware of this variation in execution times and try to use faster executing instructions when possible. This warning particularly applies to the extra shift instructions (RLC, RRC, RL, RR, SRA, SRL) and to instructions involving the index registers IX and IY.

**FASTER AND  
SLOWER  
EXECUTING  
INSTRUCTIONS**

**There are a few minor incompatibilities between the 8080A and the Z80.** These are:

**8080A/Z80  
INCOMPATIBILITIES**

- 1) The Z80 uses the P (or P/O) flag to indicate two's complement overflow after arithmetic operations. The 8080A always uses this flag for parity.
- 2) The Z80 and 8080A execute the DAA instruction differently. On the Z80, this instruction will correct decimal subtraction as well as decimal addition. On the 8080A, it will correct only decimal addition.
- 3) The Z80 rotate instructions clear the AC flag. The 8080A rotate instructions do not affect the AC flag.

Table 3-6. Register and Flag Correspondence between Z80 and 8080A

<u>Z80 Register</u>	<u>8080A Register</u>
A	A
A'	None
B	B
B'	None
C	C
C'	None
D	D
D'	None
E	E
E'	None
F	Least Significant Half of PSW
F'	None
H	H
H'	None
I	None
IX	None
IY	None
L	L
L'	None
R	None
PC	PC
SP	SP
<u>Z80 Register Pairs</u>	<u>8080A Register Pairs</u>
BC	B
DE	D
HL	H
AF	PSW
<u>Z80 Flags</u>	<u>8080A Flags</u>
C (Carry)	C (Carry)
H (Half-Carry)	AC (Auxiliary Carry)
N (Subtract)	None
P/O (Parity/Overflow)	P (Parity)
S (Sign)	S (Sign)
Z (Zero)	Z (Zero)

The Z80 is not compatible with the extra features of the 8085 microprocessor. The codes used for RIM and SIM on the 8085 are used for relative jumps (NZ and NC) on the Z80.

**8085/Z80  
INCOMPATIBILITIES**

Instruction timings on the 8080A, 8085, and Z80 all differ. Programs that depend on precise instruction timings will therefore execute properly only on the processor for which they were written.

**TIMING  
INCOMPATIBILITIES**

The N flag on the Z80 occupies bit 2 of the F register; the corresponding bit in the Processor Status Word of the 8080A is always a logic '1'.

Table 3-7 Correspondence between 8080A and Z80 Mnemonics

8080A Mnemonic		Z80 Mnemonic		8080A Mnemonic		Z80 Mnemonic	
ACI	data	ADC	A,data	LHLD	addr	LD	HL,(addr)
ADC	reg or M	ADC	A,reg or (HL)	LXI	rp,data16	LD	rp,data16
ADD	reg or M	ADD	A,reg or (HL)	MOV	reg,reg or M	LD	reg,reg or (HL)
ADI	data	ADD	A,data	MOV	reg or M,reg	LD	reg or (HL),reg
ANA	reg or M	AND	reg or (HL)	MVI	reg or M,data	LD	reg or (HL),data
ANI	data	AND	data	NOP		NOP	
CALL	addr	CALL	addr	ORA	reg or M	OR	reg or (HL)
CC	addr	CALL	C,addr	ORI	data	OR	data
CM	addr	CALL	M,addr	OUT	port	OUT	(port),A
CMA		CPL		PCHL		JP	(HL)
CMC		CCF		POP	pr	POP	pr
CMP	reg or M	CP	reg or (HL)	PUSH	pr	PUSH	pr
CNC	addr	CALL	NC,addr	RAL		RLA	
CNZ	addr	CALL	NZ,addr	RAR		RRA	
CP	addr	CALL	P,addr	RC		RET	C
CPE	addr	CALL	PE,addr	RET		RET	
CPI	data	CP	data	RLC		RLCA	
CPO	addr	CALL	PO,addr	RM		RET	M
CZ	addr	CALL	Z,addr	RNC		RET	NC
DAA		DAA		RNZ		RET	NZ
DAD	rp	ADD	HL,rp	RP		RET	P
DCR	reg or M	DEC	reg or (HL)	RPE		RET	PE
DCX	rp	DEC	rp	RPO		RET	PO
DI		DI		RRC		RRCA	
EI		EI		RST	n	RST	n
HLT		HALT		RZ		RET	Z
IN	port	IN	A,(port)	SBB	reg or M	SBC	A,reg or (HL)
INR	reg or M	INC	reg or (HL)	SBI	data	SBC	A,data
INX	rp	INC	rp	SHLD	addr	LD	(addr),HL
JC	addr	JP	C,addr	SPHL		LD	SP,HL
JM	addr	JP	M,addr	STA	addr	LD	(addr),A
JMP	addr	JP	addr	STAX	B or D	LD	(BC) or (DE),A
JNC	addr	JP	NC,addr	STC		SCF	
JP	addr	JP	P,addr	SUB	reg or M	SUB	reg or (HL)
JNZ	addr	JP	NZ,addr	SUI	data	SUB	data
JPE	addr	JP	PE,addr	XCHG		EX	DE,HL
JPO	addr	JP	PO,addr	XRA	reg or M	XOR	reg or (HL)
JZ	addr	JP	Z,addr	XRI	data	XOR	data
LDA	addr	LD	A,(addr)	XTHL		EX	(SP),HL
LDAX	B or D	LD	A,(BC) or (DE)				

Table 3-8. Correspondence between Z80 and 8080A Mnemonics

Z80 Mnemonic		8080A Mnemonic		Z80 Mnemonic		8080A Mnemonic	
ADC	A,data	ACI	data	INC	rp	INX	rp
ADC	A,(HL)	ADC	M	INC	xy	---	---
ADC	A,reg	ADC	reg	INC	(xy + disp)	---	---
ADC	A,(xy + disp)	---	---	IND	---	---	---
ADC	HL,rp	---	---	INDR	---	---	---
ADD	A,data	ADI	data	INI	---	---	---
ADD	A,(HL)	ADD	M	INIR	---	---	---
ADD	A,reg	ADD	reg	JP	addr	JMP	addr
ADD	A,(xy + disp)	---	---	JP	C,addr	JC	addr
ADD	HL,rp	DAD	rp	JP	(HL)	PCHL	---
ADD	IX,pp	---	---	JP	M,addr	JM	addr
ADD	IY,rr	---	---	JP	NC,addr	JNC	addr
AND	data	ANI	data	JP	NZ,addr	JNZ	addr
AND	(HL)	ANA	M	JP	P,addr	JP	addr
AND	reg	ANA	reg	JP	PE,addr	JPE	addr
AND	(xy + disp)	---	---	JP	PO,addr	JPO	addr
BIT	b,(HL)	---	---	JP	Z,addr	JZ	addr
BIT	b,reg	---	---	JP	xy	---	---
BIT	b,(xy + disp)	---	---	JR	C,disp	---	---
CALL	addr	CALL	addr	JR	disp	---	---
CALL	C,addr	CC	addr	JR	NC,disp	---	---
CALL	M,addr	CM	addr	JR	NZ,disp	---	---
CALL	NC,addr	CNC	addr	JR	Z,disp	---	---
CALL	NZ,addr	CNZ	addr	LD	A,(addr)	LDA	addr
CALL	P,addr	CP	addr	LD	A,(BC) or (DE)	LDAX	B or D
CALL	PE,addr	CPE	addr	LD	A,I	---	---
CALL	PO,addr	CPO	addr	LD	A,R	---	---
CALL	Z,addr	CZ	addr	LD	(addr),A	STA	addr
CCF	---	CMC	---	LD	(addr),BC or DE	---	---
CP	data	CPI	data	LD	(addr),HL	SHLD	addr
CP	(HL)	CMP	M	LD	(addr),SP	---	---
CP	reg	CMP	reg	LD	(addr),xy	---	---
CP	(xy + disp)	---	---	LD	(BC) or (DE),A	STAX	B or D
CPD	---	---	---	LD	BC or DE,(addr)	---	---
CPDR	---	---	---	LD	HL,(addr)	LHLD	addr
CPI	---	---	---	LD	(HL),data	MVI	M,data
CPIR	---	---	---	LD	(HL),reg	MOV	M,reg
CPL	---	CMA	---	LD	I,A	---	---
DAA	---	DAA	---	LD	R,A	---	---
DEC	(HL)	DCR	M	LD	reg,data	MVI	reg,data
DEC	reg	DCR	reg	LD	reg,(HL)	MOV	reg,M
DEC	rp	DCX	rp	LD	reg,reg	MOV	reg,reg
DEC	xy	---	---	LD	reg,(xy + disp)	---	---
DEC	(xy + disp)	---	---	LD	rp,data16	LXI	rp,data16
DI	---	DI	---	LD	SP,(addr)	---	---
DJNZ	disp	---	---	LD	SP,HL	SPHL	---
EI	---	EI	---	LD	SP,xy	---	---
EX	AF,AF'	---	---	LD	xy,data16	---	---
EX	DE,HL	XCHG	---	LD	xy,(addr)	---	---
EX	(SP),HL	XTHL	---	LD	(xy + disp),data	---	---
EX	(SP),xy	---	---	LD	(xy + disp),reg	---	---
EXX	---	---	---	LDD	---	---	---
HALT	---	HLT	---	LDDR	---	---	---
IM	m	---	---	LDI	---	---	---
IN	A,(port)	IN	port	LDIR	---	---	---
IN	reg,(C)	---	---	NEG	---	---	---
INC	(HL)	INR	M	NOP	---	NOP	---
INC	reg	INR	reg	OR	data	ORI	data

--- indicates that there is no corresponding instruction.

Table 3-8. Correspondence between Z80 and 8080A Mnemonics (Continued)

Z80 Mnemonic		8080A Mnemonic		Z80 Mnemonic		8080A Mnemonic	
OR	(HL)	ORA	M	RR	(HL)	—	
OR	reg	ORA	reg	RR	reg	—	
OR	(xy + disp)	—		RR	(xy + disp)	—	
OTDR		—		RRA		RAR	
OTIR		—		RRC	(HL)	—	
OUT	(C),reg	—		RRC	reg	—	
OUT	(port),A	OUT	port	RRC	(xy + disp)	—	
OUTD		—		RRCA		RRC	
OUTI		—		RRD		—	
POP	pr	POP	pr	RST	n	RST	n
POP	xy	—		SBC	A,data	SBI	data
PUSH	pr	PUSH	pr	SBC	A,(HL)	SBB	M
PUSH	xy	—		SBC	A,reg	SBB	reg
RES	b,(HL)	—		SBC	A,(xy + disp)	—	
RES	b,reg	—		SBC	HL,rp	—	
RES	b,(xy + disp)	—		SCF		STC	
RET		RET		SET	b,(HL)	—	
RET	C	RC		SET	b,reg	—	
RET	M	RM		SET	b,(xy + disp)	—	
RET	NC	RNC		SLA	(HL)	—	
RET	NZ	RNZ		SLA	reg	—	
RET	P	RP		SLA	(xy + disp)	—	
RET	PE	RPE		SRA	(HL)	—	
RET	PO	RPO		SRA	reg	—	
RET	Z	RZ		SRA	(xy + disp)	—	
RETI		—		SRL	(HL)	—	
RETN		—		SRL	reg	—	
RL	(HL)	—		SRL	(xy + disp)	—	
RL	reg	—		SUB	data	SUI	data
RL	(xy + disp)	—		SUB	(HL)	SUB	M
RLA		RAL		SUB	reg	SUB	reg
RLC	(HL)	—		SUB	(xy + disp)	—	
RLC	reg	—		XOR	data	XRI	data
RLC	(xy + disp)	—		XOR	(HL)	XRA	M
RLCA		RLC		XOR	reg	XRA	reg
RLD		—		XOR	(xy + disp)	—	

— indicates that there is no corresponding instruction

Table 3-9. Unused 8080A Operation Codes and Their Z80 Meanings

8080A Operation Code	Z80 Use
08	EX AF,AF'
10	DJNZ disp
18	JR disp
20 (RIM on 8085)	JR NZ,disp
28	JR Z,disp
30 (SIM on 8085)	JR NC,disp
38	JR C,disp
CB	BIT, RES, RL, RLC, RR, RRC, SET, SLA, SRA, SRL
D9	EXX
DD	All instructions involving Register IX.
ED	ADC HL,rp      LD    A,i      NEG
	CPD              LD    A,R      OTDR
	CPDR            LD    (addr),rp    OTIR
	CPI              LD    I,A      OUT (C),reg
	CPIR             LD    R,A      OUTD
	IM    m          LD    rp,(addr)    OUTI
	IN    reg,(C)    LDD            RETI
	IND              LDDR          RETN
	INDR            LDI            RLD
	INI              LDIR          RRD
	INIR             SBC    HL,rp
FD	All instructions involving Register IY.