

## Appendix A

# SIC/XE Instruction Set and Addressing Modes

### Instruction Set

In the following descriptions, uppercase letters refer to specific registers. The notation  $m$  indicates a memory address,  $n$  indicates an integer between 1 and 16, and  $r1$  and  $r2$  represent register identifiers. Parentheses are used to denote the contents of a register or memory location. Thus  $A \leftarrow (m..m+2)$  specifies that the contents of the memory locations  $m$  through  $m+2$  are loaded into register  $A$ ;  $m..m+2 \leftarrow (A)$  specifies that the contents of register  $A$  are stored in the word that begins at address  $m$ .

The letters in the Notes column have the following meanings:

- P Privileged instruction
- X Instruction available only on XE version
- F Floating-point instruction
- C Condition code CC set to indicate result of operation ( $<$ ,  $=$ , or  $>$ )

The Format column indicates which SIC/XE instruction format is to be used in assembling each instruction; 3/4 means that either Format 3 or Format 4 can be used. All instructions for the standard version of SIC are assembled using the format described in Section 1.3.1 (which is compatible with Format 3). Instruction subfields that are not required, such as the address field for an RSUB instruction, are set to zero.

+JSUB  
ADD m  
+ADD ALPHA

S, X  
ADDR S, X  
ADDR X, S

COMP K300

→ JSUB READ  
→ LDA ZERO  
PC

Mnemonic	Format	Opcode	Effect	Notes
ADD m	3/4	18	A $\leftarrow$ (A) + (m..m+2)	
ADDF m	3/4	58	F $\leftarrow$ (F) + (m..m+5)	X F
ADDR r1,r2	2	90	r2 $\leftarrow$ (r2) + (r1)	X
AND m	3/4	40	A $\leftarrow$ (A) & (m..m+2)	
CLEAR r1	2	B4	r1 $\leftarrow$ 0	X
COMP m	3/4	28	(A) : (m..m+2)	C
COMPF m	3/4	88	(F) : (m..m+5)	X F C
COMPR r1,r2	2	A0	(r1) : (r2)	X C
DIV m	3/4	24	A $\leftarrow$ (A) / (m..m+2)	
DIVF m	3/4	64	F $\leftarrow$ (F) / (m..m+5)	X F
DIVR r1,r2	2	9C	r2 $\leftarrow$ (r2) / (r1)	X
FIX	1	C4	A $\leftarrow$ (F) [convert to integer]	X F
FLOAT	1	C0	F $\leftarrow$ (A) [convert to floating]	X F
HIO	1	F4	Halt I/O channel number (A)	P X
J m	3/4	3C	PC $\leftarrow$ m	
JEQ m	3/4	30	PC $\leftarrow$ m if CC set to =	
JGT m	3/4	34	PC $\leftarrow$ m if CC set to >	
JLT m	3/4	38	PC $\leftarrow$ m if CC set to <	
JSUB m	3/4	48	L $\leftarrow$ (PC); PC $\leftarrow$ m	
LDA m	3/4	00	A $\leftarrow$ (m..m+2)	
LDB m	3/4	68	B $\leftarrow$ (m..m+2)	X
LDCH m	3/4	50	A [rightmost byte] $\leftarrow$ (m)	
LDF m	3/4	70	F $\leftarrow$ (m..m+5)	X F
LDL m	3/4	08	L $\leftarrow$ (m..m+2)	
LDS m	3/4	6C	S $\leftarrow$ (m..m+2)	X
LDT m	3/4	74	T $\leftarrow$ (m..m+2)	X
LDX m	3/4	04	X $\leftarrow$ (m..m+2)	
LPS m	3/4	D0	Load processor status from information beginning at address m (see Section 6.2.1)	P X
MUL m	3/4	20	A $\leftarrow$ (A) * (m..m+2)	

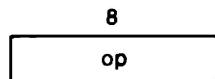
Mnemonic	Format	Opcode	Effect	Notes
MULF m	3/4	60	$F \leftarrow (F) * (m..m+5)$	X F
MULR r1, r2	2	98	$r2 \leftarrow (r2) * (r1)$	X
NORM	1	C8	$F \leftarrow (F)$ [normalized]	X F
OR m	3/4	44	$A \leftarrow (A) \mid (m..m+2)$	
RD m	3/4	D8	$A$ [rightmost byte] $\leftarrow$ data from device specified by (m)	P
RMO r1,r2	2	AC	$r2 \leftarrow (r1)$	X
RSUB	3/4	4C	$PC \leftarrow (L)$	
SHIFTL r1,n	2	A4	$r1 \leftarrow (r1)$ ; left circular shift n bits. {In assembled instruction, $r2 = n-1$ }	X
SHIFTR r1,n	2	A8	$r1 \leftarrow (r1)$ ; right shift n bits, with vacated bit positions set equal to leftmost bit of (r1). {In assembled instruction, $r2 = n-1$ }	X
SIO	1	F0	Start I/O channel number (A); address of channel program is given by (S)	P X
SSK m	3/4	EC	Protection key for address m $\leftarrow (A)$ (see Section 6.2.4)	P X
STA m	3/4	0C	$m..m+2 \leftarrow (A)$	
STB m	3/4	78	$m..m+2 \leftarrow (B)$	X
STCH m	3/4	54	$m \leftarrow (A)$ [rightmost byte]	
STF m	3/4	80	$m..m+5 \leftarrow (F)$	X F
STI m	3/4	D4	Interval timer value $\leftarrow (m..m+2)$ (see Section 6.2.1)	P X
STL m	3/4	14	$m..m+2 \leftarrow (L)$	
STS m	3/4	7C	$m..m+2 \leftarrow (S)$	X
STSW m	3/4	E8	$m..m+2 \leftarrow (SW)$	P
STT m	3/4	84	$m..m+2 \leftarrow (T)$	X
STX m	3/4	10	$m..m+2 \leftarrow (X)$	
SUB m	3/4	1C	$A \leftarrow (A) - (m..m+2)$	
SUBF m	3/4	5C	$F \leftarrow (F) - (m..m+5)$	X F

**TIX ELEVEN**

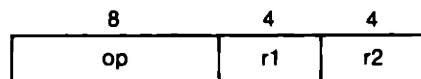
Mnemonic	Format	Opcode	Effect	Notes
SUBR r1,r2	2	94	$r2 \leftarrow (r2) - (r1)$	X
SVC n	2	B0	Generate SVC interrupt. {In assembled instruction, r1 = n}	X
TD m	3/4	E0	Test device specified by (m)	P C
TIO	1	F8	Test I/O channel number (A)	P X C
TIX m	3/4	2C	$X \leftarrow (X) + 1; (X): (m..m+2)$	C
TIXR r1	2	B8	$X \leftarrow (X) + 1; (X): (r1)$	X C
WD m	3/4	DC	Device specified by (m) $\leftarrow (A)$ [rightmost byte]	P

## Instruction Formats

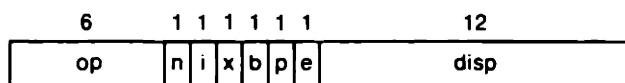
Format 1 (1 byte):



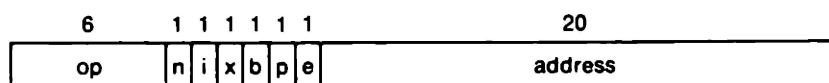
Format 2 (2 bytes):



Format 3 (3 bytes):



Format 4 (4 bytes):



## Addressing Modes

The following addressing modes apply to Format 3 and 4 instructions. Combinations of addressing bits not included in this table are treated as errors by the machine. In the description of assembler language notation, *c* indicates a constant between 0 and 4095 (or a memory address known to be in this range); *m* indicates a memory address or a constant value larger than 4095. Further information can be found in Section 1.3.2.

The letters in the Notes column have the following meanings:

- 4 Format 4 instruction
- D Direct-addressing instruction
- A Assembler selects either program-counter relative or base-relative mode
- S Compatible with instruction format for standard SIC machine.  
Operand value can be between 0 and 32,767 (see Section 1.3.2 for details).

Addressing type	Flag bits n i x b p e	Assembler language notation	Calculation of target address TA	Operand	Notes
Simple	1 1 0 0 0 0	op c	disp	(TA)	D
	1 1 0 0 0 1	+op m	addr	(TA)	4 D
	1 1 0 0 1 0	op m	(PC) + disp	(TA)	A
	1 1 0 1 0 0	op m	(B) + disp	(TA)	A
	1 1 1 0 0 0	op c,X	disp + (X)	(TA)	D
	1 1 1 0 0 1	+op m,X	addr + (X)	(TA)	4 D
	1 1 1 0 1 0	op m,X	(PC) + disp + (X)	(TA)	A
	1 1 1 1 0 0	op m,X	(B) + disp + (X)	(TA)	A
	0 0 0 - - -	op m	b/p/e/disp	(TA)	D S
	0 0 1 - - -	op m,X	b/p/e/disp + (X)	(TA)	D S
Indirect	1 0 0 0 0 0	op @c	disp	((TA))	D
	1 0 0 0 0 1	+op @m	addr	((TA))	4 D
	1 0 0 0 1 0	op @m	(PC) + disp	((TA))	A
	1 0 0 1 0 0	op @m	(B) + disp	((TA))	A
Immediate	0 1 0 0 0 0	op #c	disp	TA	D
	0 1 0 0 0 1	+op #m	addr	TA	4 D
	0 1 0 0 1 0	op #m	(PC) + disp	TA	A
	0 1 0 1 0 0	op #m	(B) + disp	TA	A

