

## Appendix B

# ASCII Character Codes

Hex code	ASCII character	Hex Code	ASCII character	Hex code	ASCII character	Hex code	ASCII character
00	NUL	20	SP	40	@	60	`
01	SOH	21	!	41	A	61	a
02	STX	22	"	42	B	62	b
03	ETX	23	#	43	C	63	c
04	EOT	24	\$	44	D	64	d
05	ENQ	25	%	45	E	65	e
06	ACK	26	&	46	F	66	f
07	BEL	27	'	47	G	67	g
08	BS	28	(	48	H	68	h
09	HT	29	)	49	I	69	i
0A	LF	2A	*	4A	J	6A	j
0B	VT	2B	+	4B	K	6B	k
0C	FF	2C	,	4C	L	6C	l
0D	CR	2D	-	4D	M	6D	m
0E	SO	2E	.	4E	N	6E	n
0F	SI	2F	/	4F	O	6F	o
10	DLE	30	0	50	P	70	p
11	DC1	31	1	51	Q	71	q
12	DC2	32	2	52	R	72	r
13	DC3	33	3	53	S	73	s
14	DC4	34	4	54	T	74	t
15	NAK	35	5	55	U	75	u
16	SYN	36	6	56	V	76	v
17	ETB	37	7	57	W	77	w
18	CAN	38	8	58	X	78	x
19	EM	39	9	59	Y	79	y
1A	SUB	3A	:	5A	Z	7A	z
1B	ESC	3B	;	5B	[	7B	{
1C	FS	3C	<	5C	\	7C	
1D	GS	3D	=	5D	]	7D	}
1E	RS	3E	>	5E	^	7E	~
1F	US	3F	?	5F	_	7F	DEL

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 <	
<u>JSUB m</u>	3/4	48	$L \leftarrow (PC); PC \leftarrow m$	X
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	$A \leftarrow (A) * (m..m+2)$	

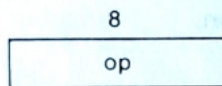


Mnemonic	Format	Opcode	Effect	Notes
MULF m	3/4	60	$F \leftarrow (F) * (m..m+5)$	X F
MULR r1,r2	2	98	$r2 \leftarrow (r2) * (r1)$	X
NORM	1	C8	$F \leftarrow (F)$ [normalized]	X F
OR m	3/4	44	$A \leftarrow (A)   (m..m+2)$	
RD m	3/4	D8	A [rightmost byte] $\leftarrow$ data from device specified by (m)	P
RMO 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

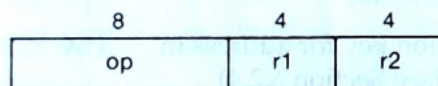
Mnemonic	Format	Opcode	Effect	Notes
SUBR r1,r2	2	94	$r2 \leftarrow (r2) - (r1)$	X
SVC n	2	B0	Generate SVC interrupt. (In assembled instruction, $r1 = n$ )	X
TD m	3/4	E0	Test device specified by (m)	P C
TIO	1	F8	Test I/O channel number (A)	P X C
TIX m	3/4	2C	$X \leftarrow (X) + 1$ ; (X): (m..m+2)	C
TIXR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

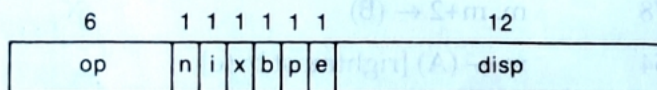
### Format 1 (1 byte):



### Format 2 (2 bytes):



### Format 3 (3 bytes):



### Format 4 (4 bytes):

