Instruction Specifier	Mnemonic	Instruction	Addressing Modes	Status Bits
0000 0000	STOP	Stop execution	U	
0000 0001	RETTR	Return from trap	U	
0000 0010	MOVSPA	Move SP to A	U	
0000 0011	MOVFLGA	Move NZVC flags to A	U	
0000 010a	BR	Branch unconditional	i, x	
0000 011a	BRLE	Branch if less than or equal to	i, x	
0000 100a	BRLT	Branch if less than	i, x	
0000 101a	BREQ	Branch if equal to	i, x	
0000 110a	BRNE	Branch if not equal to	i, x	
0000 111a	BRGE	Branch if greater than or equal to	i, x	
0001 000a	BRGT	Branch if greater than	i, x	
0001 001a	BRV	Branch if V	i, x	
0001 010a	BRC	Branch if C	i, x	
0001 011a	CALL	Call subroutine	i, x	
0001 100r	NOTr	Bitwise invert r	U	NZ
0001 101r	NEGr	Negate r	U	NZV
0001 110r	ASLr	Arithmetic shift left r	U	NZVC
0001 111r	ASRr	Arithmetic shift right r	U	NZC
0010 000r	ROLr	Rotate left r	U	C
0010 001r	RORr	Rotate right r	U	C
0010 01nn	NOPn	Unary no operation trap	U	
0010 1aaa	NOP	Nonunary no operation trap	i	
0011 0aaa	DECI	Decimal input trap	d, n, s, sf, x, sx, sxf	NZV
0011 1aaa	DECO	Decimal output trap	i, d, n, s, sf, x, sx, sxf	
0100 0aaa	STRO	String output trap	d, n, sf	
0100 1aaa	CHARI	Character input	d, n, s, sf, x, sx, sxf	
0101 0aaa	CHARO	Character output	i, d, n, s, sf, x, sx, sxf	
0101 1nnn	RETn	Return from call with n local bytes	U	
0110 0aaa	ADDSP	Add to stack pointer (SP)	i, d, n, s, sf, x, sx, sxf	NZVC
0110 1aaa	SUBSP	Subtract from stack pointer (SP)	i, d, n, s, sf, x, sx, sxf	NZVC
0111 raaa	ADDr	Add to r	i, d, n, s, sf, x, sx, sxf	NZVC
1000 raaa	SUBr	Subtract from r	i, d, n, s, sf, x, sx, sxf	NZVC
1001 raaa	ANDr	Bitwise AND to r	i, d, n, s, sf, x, sx, sxf	NZ
1010 raaa	ORr	Bitwise OR to r	i, d, n, s, sf, x, sx, sxf	NZ
1011 raaa	CPr	Compare r	i, d, n, s, sf, x, sx, sxf	NZVC
1100 raaa	LDr	Load r from memory	i, d, n, s, sf, x, sx, sxf	NZ
1101 raaa	LDBYTEr	Load byte from memory	i, d, n, s, sf, x, sx, sxf	NZ
1110 raaa	STr	Store r to memory	d, n, s, sf, x, sx, sxf	
1111 raaa	STBYTEr	Store byte r to memory	d, n, s, sf, x, sx, sxf	

aaa	Addressing mode
000	Immediate
001	Direct
010	Indirect
011	Stack-relative
100	Stack-relative deferred
101	Indexed
110	Stack-indexed
111	Stack-indexed deferred

(a) The addressing-aaa field.

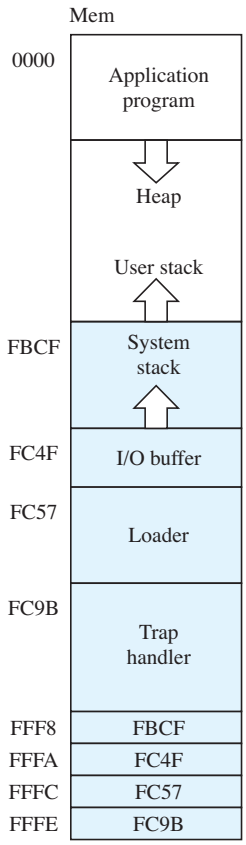
a	Addressing mode
0	Immediate
1	Indexed

(b) The addressing-a field.

r	Register
0	Accumulator, A
1	Index register, X

(c) The register-r field.

Addressing Mode	aaa	Letters	Operand
Immediate	000	i	OprndSpec
Direct	001	d	Mem [OprndSpec]
Indirect	010	n	Mem [Mem [OprndSpec]]
Stack-relative	011	s	Mem [SP + OprndSpec]
Stack-relative deferred	100	sf	Mem [Mem [SP + OprndSpec]]
Indexed	101	x	Mem [OprndSpec + X]
Stack-indexed	110	sx	Mem [SP + OprndSpec + X]
Stack-indexed deferred	111	sxf	Mem [Mem [SP + OprndSpec] + X]



Mnemonic	Register transfer language specification
STOP	Stop execution
RETTR	NZVC ← Mem[SP] ⟨4..7⟩ ; A ← Mem[SP + 1] ; X ← Mem[SP + 3] ; PC ← Mem[SP + 5] ; SP ← Mem[SP + 7]
MOVSPA	A ← SP
MOVFLAGA	A ⟨0..11⟩ ← 0 , A⟨12..15⟩ ← NZVC
BR	PC ← Oprnd
BRLE	N = 1 ∨ Z = 1 ⇒ PC ← Oprnd
BRLT	N = 1 ⇒ PC ← Oprnd
BREQ	Z = 1 ⇒ PC ← Oprnd
BRNE	Z = 0 ⇒ PC ← Oprnd
BRGE	N = 0 ⇒ PC ← Oprnd
BRGT	N = 0 ∧ Z = 0 ⇒ PC ← Oprnd
BRV	V = 1 ⇒ PC ← Oprnd
BRC	C = 1 ⇒ PC ← Oprnd
CALL	SP ← SP − 2 ; Mem[SP]← PC ; PC ← Oprnd
NOTr	r ← ¬r ; N ← r < 0 , Z ← r = 0
NEGr	r ← −r ; N ← r < 0 , Z ← r = 0 , V ← {*overflow*}
ASLr	C ← r⟨0⟩ , r⟨0..14⟩ ← r⟨1..15⟩ , r⟨15⟩ ← 0 ; N ← r < 0 , Z ← r = 0 , V ← {*overflow*}
ASRr	C ← r⟨15⟩ , r⟨1..15⟩ ← r⟨0..14⟩ ; N ← r < 0 , Z ← r = 0
ROLr	C ← r⟨0⟩ , r⟨0..14⟩ ← r⟨1..15⟩ , r ⟨15⟩ ← C
RORr	C ← r⟨15⟩ , r⟨1..15⟩ ← r⟨0..14⟩ , r ⟨0⟩ ← C
NOPn	Trap: Unary no operation
NOP	Trap: Nonunary no operation
DECI	Trap: Oprnd ← {*decimal input*}
DECO	Trap: {*decimal output*} ← Oprnd
STRO	Trap: {*string output*} ← Oprnd
CHARI	byte Oprnd ← {*character input*}
CHARO	{*character output*} ← byte Oprnd
RETn	SP ← SP + n ; PC ← Mem[SP] ; SP ← SP + 2
ADDSP	SP ← SP + Oprnd ; N ← SP < 0 , Z ← SP = 0 , V ← {*overflow*} , C ← {*carry*}
SUBSP	SP ← SP – Oprnd ; N ← SP < 0 , Z ← SP = 0 , V ← {*overflow*} , C ← {*carry*}
ADDr	r ← r + Oprnd ; N ← r < 0 , Z ← r = 0 , V ← {*overflow*} , C ← {*carry*}
SUBr	r ← r – Oprnd ; N ← r < 0 , Z ← r = 0 , V ← {*overflow*} , C ← {*carry*}
ANDr	r ← r ∧ Oprnd ; N ← r < 0 , Z ← r = 0
ORr	r ← r ∨ Oprnd ; N ← r < 0 , Z ← r = 0
CPr	T ← r – Oprnd ; N ←T < 0, Z ← T = 0 , V ← { *overflow*} , C ← {*carry*}
LDr	r ← Oprnd ; N ← r < 0 , Z ← r = 0
LDBYTEr	r⟨8..15⟩ ← byte Oprnd ; N ← r < 0 , Z ← r = 0
STr	Oprnd ← r
STBYTEr	byte Oprnd ← r⟨8..15⟩
Trap	T ← Mem[FFFA] ; Mem[T – 1] ← IR ; Mem[T – 3] ← SP ; Mem[T – 5] ← PC ; Mem[T – 7] ← X ; Mem[T – 9] ← A ; Mem[T – 10]⟨4..7⟩ ← NZVC ; SP ← T – 10 ; PC ← Mem[FFFE]

Char	Bin	Hex	Char	Bin	Hex	Char	Bin	Hex	Char	Bin	Hex
NUL	000 0000	00	SP	010 0000	20	@	100 0000	40	`	110 0000	60
SOH	000 0001	01	!	010 0001	21	A	100 0001	41	a	110 0001	61
STX	000 0010	02	"	010 0010	22	B	100 0010	42	b	110 0010	62
ETX	000 0011	03	#	010 0011	23	C	100 0011	43	c	110 0011	63
EOT	000 0100	04	$	010 0100	24	D	100 0100	44	d	110 0100	64
ENQ	000 0101	05	%	010 0101	25	E	100 0101	45	e	110 0101	65
ACK	000 0110	06	&	010 0110	26	F	100 0110	46	f	110 0110	66
BEL	000 0111	07	'	010 0111	27	G	100 0111	47	g	110 0111	67
BS	000 1000	08	(	010 1000	28	H	100 1000	48	h	110 1000	68
HT	000 1001	09	)	010 1001	29	I	100 1001	49	i	110 1001	69
LF	000 1010	0A	*	010 1010	2A	J	100 1010	4A	j	110 1010	6A
VT	000 1011	0B	+	010 1011	2B	K	100 1011	4B	k	110 1011	6B
FF	000 1100	0C	,	010 1100	2C	L	100 1100	4C	l	110 1100	6C
CR	000 1101	0D	-	010 1101	2D	M	100 1101	4D	m	110 1101	6D
SO	000 1110	0E	.	010 1110	2E	N	100 1110	4E	n	110 1110	6E
SI	000 1111	0F	/	010 1111	2F	O	100 1111	4F	o	110 1111	6F
DLE	001 0000	10	0	011 0000	30	P	101 0000	50	p	111 0000	70
DC1	001 0001	11	1	011 0001	31	Q	101 0001	51	q	111 0001	71
DC2	001 0010	12	2	011 0010	32	R	101 0010	52	r	111 0010	72
DC3	001 0011	13	3	011 0011	33	S	101 0011	53	s	111 0011	73
DC4	001 0100	14	4	011 0100	34	T	101 0100	54	t	111 0100	74
NAK	001 0101	15	5	011 0101	35	U	101 0101	55	u	111 0101	75
SYN	001 0110	16	6	011 0110	36	V	101 0110	56	v	111 0110	76
ETB	001 0111	17	7	011 0111	37	W	101 0111	57	w	111 0111	77
CAN	001 1000	18	8	011 1000	38	X	101 1000	58	x	111 1000	78
EM	001 1001	19	9	011 1001	39	Y	101 1001	59	y	111 1001	79
SUB	001 1010	1A	:	011 1010	3A	Z	101 1010	5A	z	111 1010	7A
ESC	001 1011	1B	;	011 1011	3B	[	101 1011	5B	{	111 1011	7B
FS	001 1100	1C	<	011 1100	3C	\	101 1100	5C	|	111 1100	7C
GS	001 1101	1D	=	011 1101	3D	]	101 1101	5D	}	111 1101	7D
RS	001 1110	1E	>	011 1110	3E	^	101 1110	5E	~	111 1110	7E
US	001 1111	1F	?	011 1111	3F	_	101 1111	5F	DEL	111 1111	7F

Abbreviations for Control Characters

NUL null, or all zeros
SOH start of heading
STX start of text
ETX end of text

EOT end of transmission
ENQ enquiry
ACK acknowledge
BEL bell

BS backspace
HT horizontal tabulation
LF line feed
VT vertical tabulation

FF form feed
CR carriage return
SO shift out
SI shift in

DLE data link escape
DC1 device control 1
DC2 device control 2
DC3 device control 3

DC4 device control 4
NAK negative acknowledge
SYN synchronous idle
ETB end of transmission block

CAN cancel
EM end of medium
SUB substitute
ESC escape

FS file separator
GS group separator
RS record separator
US unit separator

SP space
DEL delete