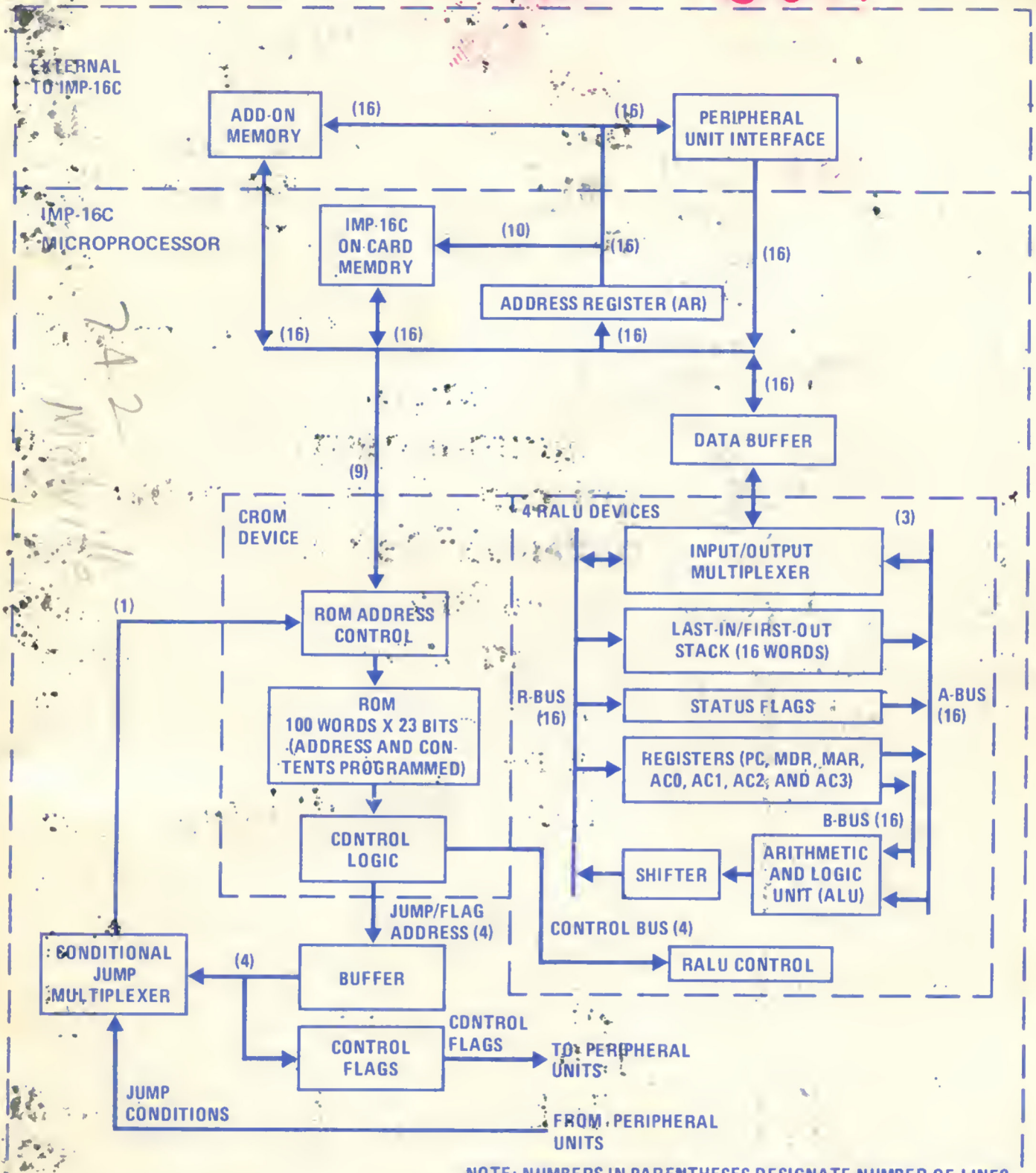
IMP-16™ INSTRUCTION GUIDE



NOTE: NUMBERS IN PARENTHESES DESIGNATE NUMBER OF LINES.

PUB. NO. 4200056

NATIONAL SEMICONDUCTOR CORPORATION
2900 SEMICONDUCTOR DRIVE
SANTA CLARA, CALIFORNIA 95051

INSTRUCTION SET (NUMERICAL ORDER)

OP CODE BASE	MNEMONIC AND ASSEMBLER FORMAT		EXECUTION CYCLES	MEMORY CYCLES		COMMAND TYPE	FORMAT GROUP
				READ	WRITE		
0000	HALT		–	1	–	BASIC	8
0080	PUSHF		4	1	–		8
0100	RTI	[+IMMED]	5	1	–		8
0200	RTS	[+IMMED]	4	1	–		8
0280	PULLF		5	1	–		8
0300	JSRP	+IMMED	8	3	–	EXTENDED	11
0380	JSRI	ADDRESS	4	1	–	BASIC	8
0400	RIN	+IMMED	7	1	–		8
0480	MPY	ADDRESS [(xr)]	106 TO 122	3	–	EXTENDED	9
0490	DIV	ADDRESS [(xr)]	125 TO 159	3	–		9
04A0	DADD	ADDRESS [(xr)]	12	4	–		9
04B0	DSUB	ADDRESS [(xr)]	12	4	–		9
04C0	LLB/LRB/LDB	ADDRESS [(xr)]	20 (LEFT) 12 (RIGHT)	4	–		9
04D0	SLB/SRB/STB	ADDRESS [(xr)]	24 (LEFT) 17 (RIGHT)	4	1		9
0500	JMPP	IMMED 4	7	3	–		10
0510	ISCAN		9 TO 80	1	–		10
0520	JINT	SPADR	7	2	–		10
0600	ROUT	+IMMED	7	1	–	BASIC	8
0700	SETST	IMMED 4	17 TO 36	1	–	EXTENDED	10
0710	CLRST	IMMED 4	17 TO 36	1	–		10
0720	SETBIT	IMMED 4	15 TO 34	1	–		10
0730	CLRBIT	IMMED 4	15 TO 34	1	–		10
0740	SKSTF	IMMED 4	19 TO 39	1	–		10
0750	SKBIT	IMMED 4	19 TO 39	1	–		10
0760	CMPBIT	IMMED 4	15 TO 34	1	–		10
0800	SFLG	IMMED 3, [+IMMED]	4	1	–	BASIC	7
0880	PFLG	IMMED 3, [+IMMED]	4	1	–		7
1000	BOC	IMMED 4, SPADR	4 OR 5	1	–		6
2000	JMP	ADDRESS [(xr)]	3	1	–		5
2400	JMP	@ ADDRESS [(xr)]	5	2	–		5
2800	JSR	ADDRESS [(xr)]	4	1	–		5
2C00	JSR	@ ADDRESS [(xr)]	6	2	–		5
3000	RADD	SOURCE REGISTER, DESTINATION REGISTER	3	1	–		1
3080	RXCH	SOURCE REGISTER, DESTINATION REGISTER	8	1	–		1
3081	RCPY	SOURCE REGISTER, DESTINATION REGISTER	6	1	–		1
3081	NOP		6	1	–		1
3082	RXOR	SOURCE REGISTER, DESTINATION REGISTER	6	1	–		1
3083	RAND	SOURCE REGISTER, DESTINATION REGISTER	6	1	–		1
4000	PUSH	REGISTER	3	1	–		3
4400	PULL	REGISTER	3	1	–		3
4800	AISZ	REGISTER, IMMED	4 OR 5	1	–		3
4C00	LI	REGISTER, IMMED	3	1	–		3
5000	CAI	REGISTER, IMMED	3	1	–		3
5400	XCHRS	REGISTER	5	1	–		3
5800	ROL/ROR	REGISTER, IMMED	4 + 3K	1	–		2
5C00	SHL/SHR	REGISTER, IMMED	4 + 3K	1	–		2
6000	AND	ACCUMULATOR, ADDRESS [(xr)]	5	2	–		4
6800	OR	ACCUMULATOR, ADDRESS [(xr)]	5	2	–		4
7000	SKAZ	ACCUMULATOR, ADDRESS [(xr)]	6 OR 7	2	–		4
7800	ISZ	ADDRESS [(xr)]	7 OR 8	2	1		5
7C00	DSZ	ADDRESS [(xr)]	8 OR 9	2	1		5
8000	LD	REGISTER, ADDRESS [(xr)]	5	2	–		4
9000	LD	REGISTER, @ ADDRESS [(xr)]	5	3	–		4
A000	ST	REGISTER, ADDRESS [(xr)]	6	1	1		4
B000	ST	REGISTER, @ ADDRESS [(xr)]	8	2	1		4
C000	ADD	REGISTER, ADDRESS [(xr)]	5	2	–		4
D000	SUB	REGISTER, ADDRESS [(xr)]	5	2	–		4
E000	SKG	REGISTER, ADDRESS [(xr)]	8 OR 9	2			4
F000	SKNE	REGISTER, ADDRESS [(xr)]	6	2	–		4

IMP-16 ASSEMBLER ASSIGNMENT STATEMENTS

```
LABEL:    SYMBOL = EXPRESSION    ; SYMBOL IS ASSIGNED VALUE
                                 ; OF EXPRESSION
                .= 20            ; SET LOCATION COUNTER TO 20
TABLE:          . = . + 10       ; RESERVE 10 LOCATIONS FOR
                                 ; TABLE
```

ASSEMBLER DIRECTIVE STATEMENTS

ASSEMBLER FORMAT		FUNCTION
.TITLE	SYMBOL [,STRING]	NAMES LOAD MODULE
.ASECT		SPECIFIES START OF ABSOLUTE SECTION
.BSECT		SPECIFIES START OF BASE SECTOR RELOCATABLE SECTION
.TSECT		SPECIFIES START OF TOP SECTOR RELOCATABLE SECTION
.END	[ADDRESS]	PHYSICAL END OF SOURCE PROGRAM
.LIST	IMMED	LISTING OUTPUT CONTROL
.SPACE	IMMED	SPACE 'n' LINES IN OUTPUT LISTING
.PAGE	[STRING]	OUTPUT LISTING TO TOP-OF-FORM
.WORD	EXPRESSION [,EXPRESSION---]	16-BIT WORD DATA GENERATION
.ASCII	STRING [,STRING---]	DATA GENERATION FOR CHARACTER STRINGS
.GLOBL	SYMBOL [,SYMBOL---]	IDENTIFIES GLOBAL SYMBOLS
.LOCAL		ESTABLISHES NEW LOCAL SYMBOL REGION
.IF	EXPRESSION	CONDITIONAL ASSEMBLY DIRECTIVES
.ELSE		CONDITIONAL ASSEMBLY DIRECTIVES
.ENDIF		CONDITIONAL ASSEMBLY DIRECTIVES
.FORM	SYMBOL, EXP [(EXP)], ---	FIELD SPECIFICATION
.EXTD		ALLOWS EXTENDED INSTRUCTION SET TO BE USED IN ASSEMBLY

NOTATION USED IN INSTRUCTION DESCRIPTIONS

NOTATION	MEANING
ACr	DENOTES SPECIFIC WORKING REGISTER (AC0, AC1, AC2, OR AC3) WHERE 'r' IS NUMBER OF ACCUMULATOR REFERENCED IN INSTRUCTION.
AR	DENOTES ADDRESS REGISTER USED FOR ADDRESSING MEMORY OR PERIPHERAL DEVICES.
cc	DENOTES 4-BIT CONDITION CODE VALUE FOR CONDITIONAL BRANCH INSTRUCTIONS.
ctl	DENOTES 7-BIT CONTROL-FIELD VALUE FOR FLAG, INPUT/OUTPUT, AND MISCELLANEOUS INSTRUCTIONS.
CY	INDICATES THAT CARRY FLAG IS SET IF THERE IS CARRY DUE TO INSTRUCTION (EITHER ADDITION OR SUBTRACTION).
disp	STANDS FOR DISPLACEMENT VALUE AND REPRESENTS OPERAND IN NONMEMORY REFERENCE INSTRUCTION OR ADDRESS FIELD IN MEMORY REFERENCE INSTRUCTION. IT IS 8-BIT, SIGNED TWOS-COMPLEMENT NUMBER EXCEPT WHEN BASE PAGE IS REFERENCED; IN LATTER CASE, IT IS UNSIGNED.

NOTATION USED IN INSTRUCTION DESCRIPTIONS (con't.)

NOTATION	MEANING
dr	DENOTES NUMBER OF DESTINATION WORKING REGISTER THAT IS SPECIFIED IN INSTRUCTION-WORD FIELD. WORKING REGISTER IS LIMITED TO ONE OF FOUR: AC0, AC1, AC2, OR AC3.
EA	DENOTES EFFECTIVE ADDRESS SPECIFIED BY INSTRUCTION DIRECTLY, INDIRECTLY, OR BY INDEXING. CONTENTS OF EFFECTIVE ADDRESS ARE USED DURING EXECUTION OF INSTRUCTION.
fc	DENOTES NUMBER OF REFERENCED FLAG.
INTEN	DENOTES INTERRUPT ENABLE CONTROL FLAG.
IOREG	DENOTES INPUT/OUTPUT REGISTER IN PERIPHERAL DEVICE.
L	DENOTES 1-BIT LINK (L) FLAG.
OV	INDICATES THAT OVERFLOW FLAG IS SET IF THERE IS OVERFLOW DUE TO INSTRUCTION (EITHER ADDITION OR SUBTRACTION).
PC	DENOTES PROGRAM COUNTER. DURING ADDRESS FORMATION, IT IS INCREMENTED BY 1 TO CONTAIN ADDRESS 1 GREATER THAN THAT OF INSTRUCTION BEING EXECUTED.
r	DENOTES NUMBER OF WORKING REGISTER THAT IS SPECIFIED IN INSTRUCTION-WORD FIELD. WORKING REGISTER IS LIMITED TO ONE OF FOUR: AC0, AC1, AC2, OR AC3.
SEL	DENOTES SELECT CONTROL FLAG. IT IS USED TO SELECT CARRY OR OVERFLOW FOR OUTPUT ON CARRY AND OVERFLOW (CYOV) LINE OF CPU AND TO INCLUDE LINK BIT (L) IN SHIFT OPERATIONS.
SPADR	SPECIAL ADDRESS.
STK	TOP WORD OF STACK.
sr	DENOTES NUMBER OF SOURCE WORKING REGISTER THAT IS SPECIFIED IN INSTRUCTION-WORD FIELD. WORKING REGISTER IS LIMITED TO ONE OF FOUR: AC0, AC1, AC2, OR AC3.
xr	WHEN NOT ZERO, THIS VALUE DESIGNATES NUMBER OF REGISTER TO BE USED IN INDEXED AND RELATIVE MEMORY-ADDRESSING MODES.
( )	DENOTES CONTENTS OF ITEM WITHIN PARENTHESES. (ACr) IS READ AS "THE CONTENTS OF ACr." (EA) IS READ AS "THE CONTENTS OF EA."
[ ]	DENOTES OPTIONAL ENTRY.
~	INDICATES LOGICAL COMPLEMENT (ONES-COMPLEMENT) OF VALUE ON RIGHT-HAND SIDE OF ~.
→	MEANS "REPLACES."
←	MEANS "IS REPLACED BY."
@	APPEARING IN OPERAND FIELD OF INSTRUCTION, DENOTES INDIRECT ADDRESSING.
∧	DENOTES **AND** OPERATION.
∨	DENOTES **OR** OPERATION.
∇	DENOTES **EXCLUSIVE OR** OPERATION.

BASIC INSTRUCTION SET

1. REGISTER TO REGISTER — DATA MOVEMENT

MNEMONIC	OPERATION	EQUATION
RADD	REGISTER ADD	(ACdr) ← (ACsr) + (ACdr), OV, CY
RXCH	REGISTER EXCHANGE	(ACsr) ← (ACdr), (ACdr) ← (ACsr)
RCPY	REGISTER COPY	(ACdr) ← (ACsr)
RXOR	REGISTER EXCLUSIVE OR	(ACdr) ← (ACdr) $\bar{\vee}$ (ACsr)
RAND	REGISTER AND	(ACdr) ← (ACdr) $\wedge$ (ACsr)
NOP	NO OPERATION	

OP CODE BASE	BASE CODE MODIFIER	REMARKS
3000 3080 3081 3082 3083	OP CODE = BASE + sr + dr sr: 0000 = AC0, 0400 = AC1, 0800 = AC2, 0C00 = AC3 dr: 0000 = AC0, 0100 = AC1, 0200 = AC2, 0300 = AC3	
3081		SIMULATED RCPY COMMAND WHERE dr = 0, sr = 0

2. REGISTER REFERENCE — SHIFT/ROTATE

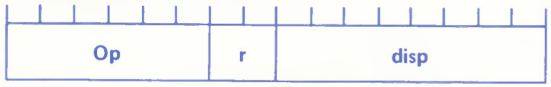
MNEMONIC	OPERATION	EQUATION	
		SEL = 0	SEL = 1
ROL	ROTATE LEFT (disp > 0)	$(ACr_0) \leftarrow (ACr_{15})$ $(ACr_n) \leftarrow (ACr_{n-1})$	$(ACr_0) \leftarrow (L)$ $(L) \leftarrow (ACr_{15})$ $(ACr_n) \leftarrow (ACr_{n-1})$
ROR	ROTATE RIGHT (disp < 0)	$(ACr_{15}) \leftarrow (ACr_0)$ $(ACr_n) \leftarrow (ACr_{n+1})$	$(ACr_{15}) \leftarrow (L)$ $(L) \leftarrow (ACr_0)$ $(ACr_n) \leftarrow (ACr_{n+1})$
SHL	SHIFT LEFT (disp > 0)	$(ACr_n) \leftarrow (ACr_{n-1})$ $(ACr_0) \leftarrow 0$	$(L) \leftarrow (ACr_{15})$ $(ACr_n) \leftarrow (ACr_{n-1})$ $(ACr_0) \leftarrow 0$
SHR	SHIFT RIGHT (disp < 0)	$(ACr_{15}) \leftarrow 0$ $(ACr_n) \leftarrow (ACr_{n+1})$	$(L) \leftarrow 0$ $(ACr_{15}) \leftarrow (L)$ $(ACr_n) \leftarrow (ACr_{n+1})$

OP CODE BASE	BASE CODE MODIFIER	REMARKS
5800	OP CODE = BASE + r + disp r: 0000 = AC0, 0100 = AC1, 0200 = AC2, 0300 = AC3	disp IS UNSIGNED WHEN USED IN ASSEMBLER LANGUAGE FORMAT.
5800	OP CODE = BASE + r + disp r: 0000 = AC0, 0100 = AC1, 0200 = AC2, 0300 = AC3	disp IS UNSIGNED WHEN USED IN ASSEMBLER LANGUAGE FORMAT.
5C00	OP CODE = BASE + r + disp r: 0000 = AC0, 0100 = AC1, 0200 = AC2, 0300 = AC3	disp IS UNSIGNED WHEN USED IN ASSEMBLER LANGUAGE FORMAT.
5C00	OP CODE = BASE + r + disp r: 0000 = AC0, 0100 = AC1, 0200 = AC2, 0300 = AC3	disp IS UNSIGNED WHEN USED IN ASSEMBLER LANGUAGE FORMAT.

BASIC INSTRUCTION SET

3. REGISTER REFERENCE

STACK
IMMEDIATE



MNEMONIC	OPERATION	EQUATION	OP CODE BASE
AISZ	ADD IMMEDIATE SKIP IF ZERO	(ACr) ← (ACr) + disp, OV, CY IF NEW (ACr) = 0, (PC) ← (PC) + 1	4800
LI	LOAD IMMEDIATE	(ACr) ← disp	4C00
CAI	COMPLEMENT, ADD IMMEDIATE	(ACr) ← ~ (ACr) + disp	5000
PUSH	PUSH ONTO STACK	(STK) ← (ACr)	4000
PULL	PULL FROM STACK	(ACr) ← (STK)	4400
XCHRS	EXCHANGE REGISTER AND STACK	(STK) ← (ACr), (ACr) ← (STK)	5400

BASE CODE MODIFIER: OP CODE = BASE + r + disp

r	REGISTER
0000	AC0
0100	AC1
0200	AC2
0300	AC3

REMARKS

4. MEMORY REFERENCE

LOGIC
SKIP
REG./MEM. DATA MOVEMENT
ARITHMETIC

MNEMONIC	OPERATION	EQUATION	OP CODE BASE
LD	LOAD DIRECT	(ACr) ← (EA)	8000
LD	LOAD INDIRECT	(ACr) ← ((EA))	9000
ST	STORE DIRECT	(EA) ← (ACr)	A000
ST	STORE INDIRECT	((EA)) ← (ACr)	B000
ADD	ADD	(ACr) ← (ACr) + (EA), OV, CY	C000
SUB	SUBTRACT	(ACr) ← (ACr) + ~ (EA) + 1, OV, CY	D000
SKG	SKIP IF GREATER	IF (ACr) $\rangle$ (EA), (PC) ← (PC) + 1	E000
SKNE	SKIP IF NOT EQUAL	IF (ACr) ≠ (EA), (PC) ← (PC) + 1	F000
AND	LOGICAL AND	(ACr) ← (ACr) ∧ (EA)	6000
OR	LOGICAL OR	(ACr) ← (ACr) ∨ (EA)	6800
SKAZ	SKIP IF AND IS ZERO	IF (ACr) ∧ (EA) = 0, (PC) ← (PC) + 1	7000

BASE CODE MODIFIER: OP CODE = BASE + r + xr + disp

r	REGISTER
0000	AC0
0400	AC1
0800	AC2
0C00	AC3

xr	ADDRESSING TECHNIQUE
0000	BASE SECTOR
0100	PC RELATIVE
0200	INDEXED-AC2
0300	INDEXED-AC3

REMARKS: FOR AND, OR, SKAZ, ONLY REGISTERS AC0 AND AC1 MAY BE USED.

5. MEMORY REFERENCE

JUMP
INCREMENT
DECREMENT

MNEMONIC	OPERATION	EQUATION	OP CODE BASE
JMP	JUMP DIRECT	(PC) ← EA	2000
JMP	JUMP INDIRECT	(PC) ← (EA)	2400
JSR	JUMP SUBROUTINE DIRECT	(STK) ← (PC), (PC) ← EA	2800
JSR	JUMP SUBROUTINE INDIRECT	(STK) ← (PC), (PC) ← (EA)	2C00
ISZ	INCREMENT SKIP IF ZERO	(EA) ← (EA) + 1, IF (EA) = 0, (PC) ← (PC) + 1	7800
DSZ	DECREMENT SKIP IF ZERO	(EA) ← (EA) − 1, IF (EA) = 0, (PC) ←(PC) + 1	7C00

BASE CODE MODIFIER: OP CODE = BASE + xr + disp

xr	ADDRESSING TECHNIQUE
0000	BASE SECTOR
0100	PC RELATIVE
0200	INDEXED-AC2
0300	INDEXED-AC3

REMARKS

BASIC INSTRUCTION SET

6. BRANCH — BRANCH ON CONDITION

MNEMONIC	OPERATION	EQUATION
BOC	BRANCH ON CONDITION	IF cc TRUE, (PC) ← (PC) + disp

OP CODE BASE	BASE CODE MODIFIER									REMARKS
1000	OP CODE = BASE + cc + disp	cc	0000	0100	0200	0300	0400	0500	0600	0700	IN CY/OV, IF SELECT FLAG (SEL) IS SET, OVERFLOW IS TESTED; IF RESET, CARRY IS TESTED.
		16L	INT	R0=0	R0⩾0	BIT0=1	BIT1=1	R0≠0	CPINT	START	
		16C	INT	R0=0	R0⩾0	BIT0=1	BIT1=1	R0≠0	CPINT	START	
		cc	0800	0900	0A00	0B00	0C00	0D00	0E00	0F00	
		16L	STFL	INEN	CY/OV	R0⩽0	POA	SEL	USER	USER	
		16C	STFL	INEN	CY/OV	R0⩽0	USER	USER	USER	USER	

7. CONTROL FLAGS — SET FLAG / PULSE FLAG

MNEMONIC	OPERATION	EQUATION
SFLG	SET FLAG	FC SET, (AR) ← ctl
PFLG	PULSE FLAG	FC PULSED, (AR) ← ctl

OP CODE BASE	BASE CODE MODIFIER									REMARKS
0800	OP CODE = BASE + fc + ctl	FLAG	8	9	10	11	12	13	14	15	CONTENTS OF AR ARE REPLACED BY ctl. FLAGS 0–7 USED BY PROCESSOR ONLY.
0880		fc	0000	0100	0200	0300	0400	0500	0600	0700	

8. I/O AND MISCELLANEOUS — INPUT/OUTPUT / MISCELLANEOUS

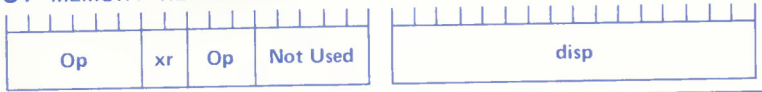
MNEMONIC	OPERATION	EQUATION
RIN	REGISTER IN	(AR) ← ctl + (AC3), (AC0) ←(IOREG)
ROUT	REGISTER OUT	(AR) ← ctl + (AC3), (IOREG) ←(AC0)
JSRI	JUMP TO SUBROUTINE IMPLIED	(STK) ← (PC), (PC) ← FF80 + ctl
RTS	RETURN FROM SUBROUTINE	(PC) ← (STK) + ctl
RTI	RETURN FROM INTERRUPT	(PC) ← (STK) + ctl, INTEN ← 1
PUSHF	PUSH STATUS FLAGS ONTO STACK	(STK) ←(STATUS FLAGS)
PULLF	PULL STATUS FLAGS FROM STACK	(STATUS FLAGS) ← (STK)
HALT	HALT	

OP CODE BASE	BASE CODE MODIFIER	REMARKS
0400	OP CODE = BASE + ctl	CONTENTS OF AR ARE REPLACED BY ctl.
0600		
0380		
0200		
0100		
0080		
0280		
0000		

EXTENDED INSTRUCTION SET

9. MEMORY REFERENCE

ARITHMETIC
LOAD
STORE



MNEMONIC	OPERATION	EQUATION
MPY	MULTIPLY	(AC0), (AC1) ← (AC1) * (EA), SEL ← 0, L UNALTERED
DIV	DIVIDE	(AC0), (AC1) ← (AC0), (AC1) ÷ (EA), OV, SEL ← 0, L UNALTERED
DADD	DOUBLE PRECISION ADD	(AC0), (AC1) ← (AC0), (AC1) + (EA), (EA) + 1, OV, SEL ← 0, CY
DSUB	DOUBLE PRECISION SUBTRACT	(AC0), (AC1) ← (AC0), (AC1) + ~[(EA), (EA) + 1] + 1, OV, CY, SEL ← 0
LDB LLB LRB	LOAD BYTE LOAD LEFT BYTE LOAD RIGHT BYTE	(LOW ORDER BYTE AC0) ← BYTE FROM (EA ÷ 2), SEL ← 0
STB SLB SRB	STORE BYTE STORE LEFT BYTE STORE RIGHT BYTE	BYTE TO (EA ÷ 2) ← (LOW ORDER BYTE OF AC0), SEL ← 0

OP CODE BASE	BASE CODE MODIFIER	REMARKS
0480 0490 04A0 04B0	OP CODE = BASE ∨ xr + disp xr: 0000 DIRECT; 0100 PC RELATIVE; 0200 INDEXED-AC2; 0300 INDEXED-AC3	AC0 AND AC1 WORKING REGISTERS—32-BIT ANSWER
04C0	OP CODE (WORD 1) = BASE + xr	IF LOW ORDER BIT OF EA IS 1, LOW ORDER BYTE IS LOADED. OTHERWISE, HIGH BYTE IS LOADED. LLB, SLB FORCES LSB TO 0. LRB, SRB FORCES LSB TO 1.
04D0	OP CODE (WORD 2) = EA X 2	

10. BIT AND STATUS FLAGS

BIT
STATUS FLAG

MNEMONIC	OPERATION	EQUATION
SETST	SET STATUS FLAG	STATUS FLAG (disp) ← 1, SEL ← 0
CLRST	CLEAR STATUS FLAG	STATUS FLAG (disp) ← 0, SEL ← 0
SKSTF	SKIP IF STATUS FLAG TRUE	IF STATUS FLAG (disp) = 1, (PC) ← (PC) + 1
SETBIT	SET BIT	(AC0disp) ← 1, SEL ← 0
CLRBIT	CLEAR BIT	(AC0disp) ← 0, SEL ← 0
CMPBIT	COMPLEMENT BIT	(AC0disp) ← ~ (AC0disp), SEL ← 0
SKBIT	SKIP IF BIT TRUE	IF (AC0disp) ← 1, (PC) ← (PC) + 1, SEL ← 0
ISCAN	INTERRUPT SCAN	IF AC1 = 0, SEL ← 0 ELSE SEL ← 0 (AC1) ← (SHIFT AC1 RIGHT UNTIL 1 SHIFTED OUT) (AC2) ← (AC2) + NUMBER OF SHIFTS (PC) ← (PC) + 1
JINT	JUMP TO LEVEL 0 INTERRUPT INDIRECT	(STK) ← (PC); (PC) ← (120_{16} + disp) INTEN ← 0
JMPP	JUMP THROUGH POINTER	(PC) ← (100_{16} + disp)

OP CODE BASE	BASE CODE MODIFIER	REMARKS
0700 0710 0740 0720 0730 0760 0750 0510 0520 0500	OP CODE = BASE + disp disp: 00 THRU 12 USER SPECIFIED; 00 THRU 13 CY (CARRY); 00 THRU 14 OV (OVERFLOW); 00 THRU 15 L (LINK)	0 ≤ disp ≤ 15

11. JUMP THROUGH POINTER

JUMP THROUGH POINTER

MNEMONIC	OPERATION	EQUATION
JSRP	JUMP TO SUBROUTINE THROUGH POINTER	(STK) ← (PC), (PC) ← (100_{16} + disp)

OP CODE BASE	BASE CODE MODIFIER	REMARKS
0300	OP CODE = BASE + disp	

INSTRUCTION SET (ALPHABETICAL ORDER)

MNEMONIC	ASSEMBLER FORMAT	OP CODE BASE	EXECUTION CYCLES	MEMORY CYCLES READ	MEMORY CYCLES WRITE	COMMAND TYPE	FORMAT GROUP
ADD	REGISTER, ADDRESS [(xr)]	C000	5	2	–	BASIC	4
AISZ	REGISTER, IMMED	4800	4 OR 5	1	–		3
AND	ACCUMULATOR, ADDRESS [(xr)]	6000	5	2	–		4
BOC	IMMED 4, SPADR	1000	4 OR 5	1	–		6
CAI	REGISTER, IMMED	5000	3	1	–		3
CLRBIT	IMMED 4	0730	15 TO 34	1	–	EXTENDED	10
CLRST	IMMED 4	0710	17 TO 36	1	–		10
CMPBIT	IMMED 4	0760	15 TO 34	1	–		10
DADD	ADDRESS [(xr)]	04A0	12	4	–		9
DIV	ADDRESS [(xr)]	0490	125 TO 159	3	–		9
DSUB	ADDRESS [(xr)]	04B0	12	4	–		9
DSZ	ADDRESS [(xr)]	7C00	8 OR 9	2	1	BASIC	5
HALT		0000	–	1	–		8
ISCAN		0510	9 TO 80	1	–	EXTENDED	10
ISZ	ADDRESS [(xr)]	7800	7 OR 8	2	1	BASIC	5
JINT	SPADR	0520	7	2	–	EXTENDED	10
JMP	ADDRESS [(xr)]	2000	3	1	–	BASIC	5
JMP	@ ADDRESS [(xr)]	2400	5	2	–		5
JMPP	IMMED 4	0500	7	3	–	EXTENDED	10
JSR	ADDRESS [(xr)]	2800	4	1	–	BASIC	5
JSR	@ ADDRESS [(xr)]	2C00	6	2	–		5
JSRI	ADDRESS	0380	4	1	–		8
JSRP	+IMMED	0300	8	3	–	EXTENDED	11
LD	REGISTER, ADDRESS [(xr)]	8000	5	2	–	BASIC	4
LD	REGISTER, @ ADDRESS [(xr)]	9000	5	3	–		4
LI	REGISTER, IMMED	4C00	3	1	–		3
LLB/LRB/LDB	ADDRESS [(xr)]	04C0	20 (LEFT) 12 (RIGHT)	4	–	EXTENDED	9
MPY	ADDRESS [(xr)]	0480	106 TO 122	3	–		9
NOP		3081	6	1	–	BASIC	1
OR	ACCUMULATOR, ADDRESS [(xr)]	6800	5	2	–		4
PFLG	IMMED 3, [+IMMED]	0880	4	1	–		7
PULL	REGISTER	4400	3	1	–		3
PULLF		0280	5	1	–		8
PUSH	REGISTER	4000	3	1	–		3
PUSHF		0080	4	1	–		8
RADD	SOURCE REGISTER, DESTINATION REGISTER	3000	3	1	–		1
RAND	SOURCE REGISTER, DESTINATION REGISTER	3083	6	1	–		1
RCPY	SOURCE REGISTER, DESTINATION REGISTER	3081	6	1	–		1
RIN	+IMMED	0400	7	1	–		8
ROL/ROR	REGISTER, IMMED	5800	4 + 3K	1	–		2
ROUT	+IMMED	0600	7	1	–		8
RTI	[+IMMED]	0100	5	1	–		8
RTS	[+IMMED]	0200	4	1	–		8
RXCH	SOURCE REGISTER, DESTINATION REGISTER	3080	8	1	–		1
RXOR	SOURCE REGISTER, DESTINATION REGISTER	3082	6	1	–		1
SETBIT	IMMED 4	0720	15 TO 34	1	–	EXTENDED	10
SETST	IMMED 4	0700	17 TO 36	1	–		10
SFLG	IMMED 3, [+IMMED]	0800	4	1	–	BASIC	7
SHL/SHR	REGISTER, IMMED	5C00	4 + 3K	2	–		2
SKAZ	ACCUMULATOR, ADDRESS [(xr)]	7000	6 OR 7	2	–		4
SKBIT	IMMED 4	0750	19 TO 39	1	–	EXTENDED	10
SKG	REGISTER, ADDRESS [(xr)]	E000	8 OR 9	2		BASIC	4
SKNE	REGISTER, ADDRSSS [(xr)]	F000	6	2	–		4
SKSTF	IMMED 4	0740	19 TO 39	1	–	EXTENDED	10
SLB/SRB/STB	ADDRESS [(xr)]	04C0	24 (LEFT) 17 (RIGHT)	4	1		9
ST	REGISTER, ADDRESS [(xr)]	A000	6	1	1	BASIC	4
ST	REGISTER, @ ADDRESS [(xr)]	B000	8	2	1		4
SUB	REGISTER, ADDRESS [(xr)]	D000	5	2	–		4
XCHRS	REGISTER	5400	5	1	–		3

ASCII/HEX CONVERSION TABLE

CHARACTER	7-BIT HEX NO.
NUL	00
SOH	01
STX	02
ETX	03
EOT	04
ENQ	05
ACK	06
BEL	07
BS	08
HT	09
LF	0A
VT	0B
FF	0C
CR	0D
SO	0E
SI	0F
DLE	10
DC1	11
DC2	12
DC3	13
DC4	14
NAK	15
SYN	16
ETB	17
CAN	18
EM	19
SUB	1A
ESC	1B
FS	1C
GS	1D
RS	1E
US	1F
SP	20

CHARACTER	7-BIT HEX NO.
!	21
"	22
#	23
$	24
%	25
&	26
'	27
(	28
)	29
*	2A
+	2B
,	2C
–	2D
.	2E
/	2F
0	30
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39
:	3A
;	3B
<	3C
=	3D
>	3E
?	3F
@	40
A	41

CHARACTER	7-BIT HEX NO.
B	42
C	43
D	44
E	45
F	46
G	47
H	48
I	49
J	4A
K	4B
L	4C
M	4D
N	4E
O	4F
P	50
Q	51
R	52
S	53
T	54
U	55
V	56
W	57
X	58
Y	59
Z	5A
[	5B
\	5C
]	5D
↑	5E
←	5F
`	60
a	61
b	62

CHARACTER	7-BIT HEX NO.
c	63
d	64
e	65
f	66
g	67
h	68
i	69
j	6A
k	6B
l	6C
m	6D
n	6E
o	6F
p	70
q	71
r	72
s	73
t	74
u	75
v	76
w	77
x	78
y	79
z	7A
	7B
	7C
ALT	7D
ESC	7E
DEL, RUBOUT	7F

CHARACTER	DEFINITION
NUL	NULL
SOH	START OF HEADING; ALSO START OF MESSAGE
STX	START OF TEXT; ALSO EOA, END OF ADDRESS
ETX	END OF TEXT; ALSO EOM, END OF MESSAGE
EOT	END OF TRANSMISSION (END)
ENQ	ENQUIRY (ENQRY); ALSO WRU
ACK	ACKNOWLEDGE. ALSO RU
BEL	RINGS THE BELL
BS	BACKSPACE
HT	HORIZONTAL TAB
LF	LINE FEED OR LINE SPACE (NEW LINE): ADVANCES PAPER TO NEXT LINE BEGINNING OF LINE
VT	VERTICAL TAB (VTAB)
FF	FORM FEED TO TOP OF NEXT PAGE (PAGE)
CR	CARRIAGE RETURN TO

CHARACTER	DEFINITION
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 (SYNC)
ETB	END OF TRANSMISSION BLOCK
CAN	CANCEL (CANCL)
EM	END OF MEDIUM
SUB	SUBSTITUTE
ESC	ESCAPE. PREFIX
FS	FILE SEPARATOR
GS	GROUP SEPARATOR
RS	RECORD SEPARATOR
US	UNIT SEPARATOR
SP	SPACE

B20M64 Printed in U.S.A.