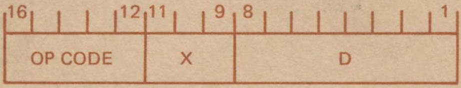
SPERRY UNIVAC

6145 ASIST REFERENCE CARD

INSTRUCTION FORMAT

SHORT FORMAT



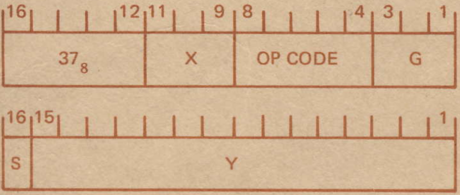
Bits 16-12	Op code of instruction to be performed
Bits 11-9	X field for address modification
Bits 8-1	D field operand or relative operand address

X Field Meaning

X	Address
0	m = P + D
1	m = X1 + D
2	m = X2 + D
3	m = X1 + X2 + D
4	m = Indirect via P + D
5	m = Indirect via X1 + D
6	m = X4 + D
7	Use D field as the operand (D is 8 bits of positive magnitude)

(D is two's complement) — applies to X = 0 through 6.

LONG FORMAT



FIRST WORD

Bits 16-12	All ones to specify LONG format
Bits 11-9	X field for address modification
Bits 8-4	Op code of instructions to be performed
Bits 3-1	G field designates certain operation

SECOND WORD

Bit 16	Sign of two's complement number in bits 15-1 when X = 7, ignored when X = 0-6
Bits 15-1	Base operand address when X = 0-6

X Field Meaning

X	Address
0	m = Y
1	m = X1 + Y
2	m = X2 + Y
3	m = X1 + X2 + Y
4	m = Indirect via Y
5	m = Indirect via X1 + Y
6	m = X4 + Y
7	m = P+1 Use full 16 bits as a signed operand (i.e., LDAL = 497 or STAL = **)

INSTRUCTION REPERTOIRE

LOAD AND STORE INSTRUCTIONS

Octal	MNE	Format	Description		
10	LDA	S/L	Load A	G = 4, 6	(m) → A
11	LDE	S/L	Load E	G = 5, 7	(m) → E
10, 11	LDXn	L	Load Index n	G = 1, 2, 3	(m) → Xn
14	DLD	S/L	Load A and E	G = 0	(m) → A, (m+1) → E
20	DST	S/L	Double Store	A and E	(A) → m, (E) → m+1
22	STA	S/L	Store A	G = 4, 6	(A) → m
23	STE	S/L	Store E	G = 5, 7	(E) → m
22, 23	STXn	L	Store Xn	G = 1, 2, 3	(Xn) → m
22, 23	STZ	L	Store Zeros	G = 0	0's → m

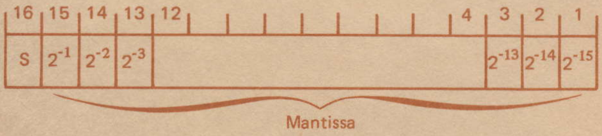
ARITHMETIC INSTRUCTIONS

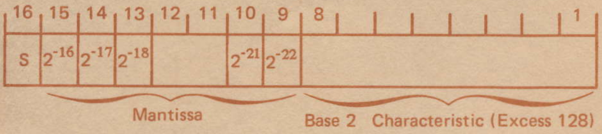
15	ADD	S/L	Add	(A) + (m) → A
24	SUB	S/L	Subtract	(A) – (m) → A
16	MPY	S/L	Multiply	(A) · (m) → AE
21	DAD	S/L	Double Add	(AE) + (m,m+1) → AE
30	DVD	S/L	Divide	(AE) ÷ (m) → E(quot)A(rem) Skip if not divide overflow

FLOATING-POINT ARITHMETIC INSTRUCTIONS

21	FAD	L	Floating-Point Add	G = 2	(AE) + (m,m+1) → AE
21	FSB	L	Floating-Point Subtract	G = 3	(AE) – (m,m+1) → AE
21	FMP	L	Floating-Point Multiply	G = 4	(AE) · (m,m+1) → AE
21	FDV	L	Floating-Point Divide	G = 5	(AE) ÷ (m,m+1) → AE

FLOATING-POINT NUMBER FORMAT




16 15 14 13 12 11 10 9 8 1
S 2^{-16} 2^{-17} 2^{-18} 2^{-21} 2^{-22}
Mantissa
Base 2 Characteristic (Excess 128)

LOGICAL INSTRUCTIONS

12	LOR	S/L	Logical OR	(E) ⊕ (m) → A
13	AND	S/L	Logical AND	(E) ⊙ (m) → A
31	XOR	S/L	Exclusive OR	$[(E) \odot (\overline{m})] \oplus [(\overline{E}) \odot (m)] \rightarrow A$

JUMP, BRANCH, SKIP INSTRUCTIONS

Octal	MNE	Format	Description	
00	JMP	S/L	Unconditional Jump	(m) → P
02	RTJ	S/L	Return Jump: short	(P+1) → m,m+1 → P,
			long	(P+2) → m,m+1 → P
03	XEC	S/L	Execute a remote instruction	
04	JTB	S/L	Jump and Test Bits	(m)+k → P
05	JGZ	S/L	Jump if (A) > 0	(m) → P if (A) > 0
06	JEZ	S/L	Jump if (A) = 0	(m) → P if (A) = 0
07	JLZ	S/L	Jump if (A) < 0	(m) → P if (A) < 0
17	IMT	S/L	Increment Memory and Test G = 1	(m+1) → m†
17	AMTn	L	Augment Memory and Test	G = 0, 2-7 (m)+G → m†
25	KAL	S/L	Skip if (A) Low	If (A) < (m), skip
26	KAQ	S/L	Skip if (A) Equal	If (A) = (m), skip
27	KAH	S/L	Skip if (A) High	If (A) > (m), skip
33	AXDn	S (two word)	Augment Index n and test If $(P+1)_{16} = 0$ then; when $(Xn)+D > (P+1)_{15\text{-}1}$, skip next location; when $(Xn)+D \leqslant (P+1)_{15\text{-}1}$, perform next instruction If $(P+1)_{16} = 1$ then; when $(Xn)+D \leqslant (P+1)_{15\text{-}1}$, skip next location; when $(Xn)+D > (P+1)_{15\text{-}1}$, perform next instruction	
32	TAS	L	Test And Set G = 4 if (m) ≠ 0 set m and execute next instruction if (m) = 0 set m and skip next location	
32	TFG	L	Test Flags G = 5 when X = 7 test bits are in Y Bit 5 Sense Switch 1 Bit 6 Sense Switch 2 Bit 7 Sense Switch 3 Bit 8 Sense Switch 4 Bit 9 Overflow Flip-Flop when all conditions tested for are false skip next location	

17
77477

SHIFT INSTRUCTIONS

34	RSA	S	Arithmetic Right Shift A	X=0, D8=0 or 1
34	RSD	S	Arithmetic Right Shift AE	X=2, D8=0 or 1
34	RLA	S	Logical Right Shift A	X=1, D8=0
34	RLD	S	Logical Right Shift AE	X=3, D8=0 or 1
34	LSA	S	Arithmetic Left Shift A	X=4, D8=0
34	LSD	S	Arithmetic Left Shift AE	X=6, D8=0
34	LLA	S	Logical Left Shift A	X=5, D8=0
34	LLD	S	Logical Left Shift AE	X=7, D8=0
34	CSA	S	Circular Left Shift A	X=5, D8=1
34	CSD	S	Circular Left Shift AE	X=7, D8=1
34	TRN	S	Transpose A and E	X=1, D8=1
34	TAL	S	Tally A store count in X2	X=4, D8=1
34	NRM	S	Normalize AE	X=6, D8=1
34	NOP	S	No Operation	X=4, D8-1=0, zero shift count

†Skip if result = 0 or causes sign change

STATUS INSTRUCTIONS

Octal	MNE	Format		Description
33	SST	S	Store Status	D2, 1=0
33	LST	S	Load Status	D2=0, D1=1

	16	15 … 1
Push Stack Pointer	0	Program Counter Register
Push Stack Pointer + 1	Accumulator Register	
Push Stack Pointer + 2	Extension Register	
Push Stack Pointer + 3	0	Index Register 1
Push Stack Pointer + 4	†	Index Register 2
Push Stack Pointer + 5	0	Index Register 4

†Overflow Flip-Flop

REGISTER MODIFICATION INSTRUCTIONS

16 … 12	11 … 6	5 4	3 … 1
35_8	DESTINATION(S)	OP	SOURCE

Source	Register	ASIST Code	Destination	Register	ASIST Code
0	Zeros	Z	Bit 11 = 1	X4	4
1	X1	1	Bit 10 = 1	X2	2
2	X2	2	Bit 9 = 1	X1	1
3	X4	4	Bit 8 = 1	A	A
4, 6	A	A	Bit 7 = 1	E	E
5, 7	E	E	Bit 6 = 1	BUFF	B

Octal	MNE	Format	Description	
35	MOV	S	Move Source to Destination(s)	OP=00
35	CMP	S	Complement Source and Move	OP=01
35	MAO	S	Add one to Source and Move	OP=10
35	NEG	S	Negate Source and Move	OP=11

WORKER BASE REGISTER AND BUFF INSTRUCTIONS

32	LWB	L	Load Worker Base 2	G=6	$(m)_{8-1} \rightarrow$ BR2
32	SWB	L	Store Worker Base 2	G=2	$(BR2) \rightarrow m_{8-1}$
32	LBU	L	Load BUFF	G=7	If X=0-6, $(m)_{16} \rightarrow$ BUFF If X=7, $(Y)_{16} \rightarrow$ BUFF
32	SBU	L	Store BUFF	G=3	If X=0-6, $(BUFF) \rightarrow m_{16}$ If X=7, $(BUFF) \rightarrow Y_{16}$

BYTE HANDLING INSTRUCTIONS

32 LDB L Load Byte

Byte Address = 1st Byte Address + $(E)_{16-2}$

$(E)_1$ = 0 Specifies Upper Byte (16-9) into A_{8-1}, 0's $\rightarrow A_{16-9}$

$(E)_1$ = 1 Specifies Lower Byte (8-1) into A_{8-1}, 0's $\rightarrow A_{16-9}$

(E) +1 → E

32 STB L Store Byte G=1

Byte Address = 1st Byte Address + $(E)_{16-2}$

$(E)_1$ = 0 Specifies A_{8-1} into Upper Byte (16-9)

$(E)_1$ = 1 Specifies A_{8-1} into Lower Byte (8-1)

(E) +1 → E

MISCELLANEOUS INSTRUCTIONS

Octal	MNE	Format	Description	
34	HLT	S	Halt if in Halt-Enabled mode	D7=0, D6=1
34	GAO	S	Get Address and Operand	D7=1, D6=0
03	SMM	L	Set Monitor Master	

PRIVILEGED INSTRUCTIONS

Privileged instructions can be executed under executive mode only.

Octal	MNE	Format	Description
33	LBS	L	Load Base 1 and Shadow Register If X = 0-6, $(m)_{5-1} \rightarrow$ BR1, $(m)_{16-11} \rightarrow$ SR If X = 7, $(Y)_{5-1} \rightarrow$ BR1, $(Y)_{16-11} \rightarrow$ SR
33	SBZ	L	Store Base Zero (BR0) $\rightarrow m_{8-7}$, $0 \rightarrow m_{16-9\ 6-1}$ G=1
01	JCI	S/L	Jump and Clear Interrupt (m) → P, clear interrupt
33	LKJ	L	Load KEY And Jump $(E)_{4-1} \rightarrow$ KEY, (m) → P G=2
33	ESS	S	Executive Save Status D2=1, D1=0
33	ELS	S	Executive Load Status D2-1=1

	16	15 … 12	11	10	9	8 … 1
Executive Push Stack Ptr. †	††	KEY	0	0	0	Base Register 2
Executive Push Stack Ptr. + 1	Shadow Register		0	0		Base Register 1
Executive Push Stack Ptr. + 2	0	Program Counter Register				
Executive Push Stack Ptr. + 3	Accumulator Register					
Executive Push Stack Ptr. + 4	Extension Register					
Executive Push Stack Ptr. + 5	0	Index Register 1				
Executive Push Stack Ptr. + 6	†††	Index Register 2				
Executive Push Stack Ptr. + 7	0	Index Register 4				

Octal	MNE	Format	Description
33	WFI	S	Wait For Interrupt (P) → P wait until interrupt then perform next instruction X=5, D=0
33	WPL	S	Write Protection LOCK $(E)_{4-1} \rightarrow$ LOCK X=4, D8-1=0 LOCK specified by $(A)_{8-1}$ (A) + 1 → A
33	RPL	S	Read Protection LOCK (LOCK) $\rightarrow E_{4-1}$ X=4, D7-2=0, D1=1 LOCK specified by $(A)_{8-1}$ (A) + 1 → A
32	PIO	S	Programmed Input/Output Example: PIO ed, iofc, x

16 … 12	11	10	9 8	7 … 1
32_8	X_4	I/O	FC	STATION ADDRESS (ED)

Bits 16-12	Operation Code 32_8 for PIO
Bit 11	When set the station address is indexed by (X4).
Bit 10	When clear Output, when set Input
Bits 9-8	Function code specifying the type of operation to be performed
Bits 7-1	Station Address of the ED

† Executive Push Stack Pointer

†† BUFF

†††Overflow Flip-Flop

EXPLANATION OF SYMBOLS

A	The A register (accumulator)
E	The E register (extension)
P	The P register (program counter)
Xn	Index register number n (n can be 1, 2, or 4)
k	Constant
n	Number of index register or with AMT instruction number to augment by
ed	External device address (station address)
fc	Function code
io	Input output
x	Indexing
m	Effective 15-bit operand address (i.e., unbased)
D	Bits 8-1 of short format instruction operand or relative operand address
X	Bits 11-9 of short format or first word of long format instruction (designates type of address modification if any)
S	Sign bit, bit 16 of second word of long format instruction
G	Bits 3-1 of first word of long format instruction, defines certain operations
(A)	Contents of A register
$(A)_{16\text{-}9}$	Contents of bits 16-9 of A register
A_1	Bit 1 of A register
$\cdot$	Multiply
$\div$	Divide
$+$	Add
$-$	Subtract
$=$	Equal to
$>$	Greater than
$<$	Less than
$\geqslant$	Greater than or equal to
$\leqslant$	Less than or equal to
$\rightarrow$	Is placed in
$\bar{n}$	Negated value of n or not n
$\oplus$	Logical OR
$\odot$	Logical AND

SPERRY UNIVAC

SERIES 60
8001 Bloomington Freeway
Minneapolis, Minnesota 55420

PP1420-002, REV. 1 Lithographed in USA 3/73

CHAR.	EBCDIC	ASCII	HOLLERITH
0	360	060	0
1	361	061	1
2	362	062	2
3	363	063	3
4	364	064	4
5	365	065	5
6	366	066	6
7	367	067	7
8	370	070	8
9	371	071	9
A	301	101	12, 1
B	302	102	12, 2
C	303	103	12, 3
D	304	104	12, 4
E	305	105	12, 5
F	306	106	12, 6
G	307	107	12, 7
H	310	110	12, 8
I	311	111	12, 9
J	321	112	11, 1
K	322	113	11, 2
L	323	114	11, 3
M	324	115	11, 4
N	325	116	11, 5
O	326	117	11, 6
P	327	120	11, 7
Q	330	121	11, 8
R	331	122	11, 9
S	342	123	0, 2
T	343	124	0, 3
U	344	125	0, 4
V	345	126	0, 5
W	346	127	0, 6
X	347	130	0, 7
Y	350	131	0, 8
Z	351	132	0, 9
+	116 (or 120)†	053	12, 6, 8 (or 12)†
—	140	055	11
*	134	052	11, 4, 8
/	141	057	0, 1
.	113	056	12, 3, 8
,	153	054	0, 3, 8
(	115 (or 154)†	050	12, 5, 8 (or 0, 4, 8)†
)	135 (or 114)†	051	11, 5, 8 (or 12, 4, 8)†
=	176 (or 173)†	075	6, 8 (or 3, 8)†
$	133	044	11, 3, 8
SPACE	100	040	
?		077	
"		042	
TAB		011	
LINE FEED		012	
CARR. RETURN		015	
RUBOUT		177	
←		137	
APOSTROPHE	175 (or 174)†	047 (or 100)†	8, 5 (or 8, 4)†

†Only for those programs which check and make them equiv.

ASSEMBLER INSTRUCTIONS

MNE	Description
CONTROL	
ORG	Origin program in absolute form
END	End of program or program segment
EOF	End of file record generation
EOJ/	End of job, return to operating system
ASSEMBLY CONTROL	
IFC	If Control
IFT	If True
IFF	If False
LIST CONTROL	
TTL	Title generation
EJT	Eject line printer page
SPC	Space line printer page k spaces
NOL	No List on line printer
LIS	List on line printer
SUBROUTINE LINKAGE	
ENT	Entry point to program
EXT	External symbol to program
CALL	Call external subroutine
SYMBOL VALUE ASSIGNMENT	
EQU	Equate variable field to location field
STORAGE ASSIGNMENT	
RES	Reserve k consecutive core locations
RZO	Reserve and zero k consecutive core locations
CMN	Common consecutive core locations
DATA GENERATION	
VAL	Value, specify values in memory
VFD	Variable field definition
SYMBOL TABLE MODIFICATION	
SAV	Save listed symbols in table
DEL	Delete listed symbols from table
CST	Clear symbol table
PST	Print symbol table
MACRO GENERATION AND CONTROL	
MAC	Macro definition
EMC	End macro
FMC	Macro listing control