UNIVAC 9400 SYSTEM

HARDWARE AND OS/4 SOFTWARE SUMMARY

SPERRY UNIVAC

UP-7625
Rev. 2

This document contains the latest information available at the time of publication. However, Sperry Univac reserves the right to modify or revise its contents. To ensure that you have the most recent information, contact your local Sperry Univac representative.

UNIVAC is a registered trademark of the Sperry Rand Corporation.

Other trademarks of the Sperry Rand Corporation include:

FASTRAND

UNISCOPE

UNISERVO

PAGEWRITER

MATED-FILM

CONTENTS

CHARACTER ①	HEXADECIMAL		80-COLUMN CARD	7-TRACK TAPE (BCDIC)②
	CONSOLE KEYBOARD (EBCDIC)	EBCDIC		
A	C1	C1	12, 1	B A 1
B	C2	C2	12, 2	B A 2
C	C3	C3	12, 3	B A 2 1
D	C4	C4	12, 4	B A 4
E	C5	C5	12, 5	B A 4 1
F	C6	C6	12, 6	B A 4 2
G	C7	C7	12, 7	B A 4 2 1
H	C8	C8	12, 8	B A 8
I	C9	C9	12, 9	B A 8 1
J	D1	D1	11, 1	B 1
K	D2	D2	11, 2	B 2
L	D3	D3	11, 3	B 2 1
M	D4	D4	11, 4	B 4
N	D5	D5	11, 5	B 4 1
O	D6	D6	11, 6	B 4 2
P	D7	D7	11, 7	B 4 2 1
Q	D8	D8	11, 8	B 8
R	D9	D9	11, 9	B 8 1
S	E2	E2	0, 2	A 2
T	E3	E3	0, 3	A 2 1
U	E4	E4	0, 4	A 4
V	E5	E5	0, 5	A 4 1
W	E6	E6	0, 6	A 4 2
X	E7	E7	0, 7	A 4 2 1
Y	E8	E8	0, 8	A 8
Z	E9	E9	0, 9	A 8 1
0	F0	F0	0	8 2
1	F1	F1	1	1
2	F2	F2	2	2
3	F3	F3	3	2 1
4	F4	F4	4	4
5	F5	F5	5	4 1
6	F6	F6	6	4 2
7	F7	F7	7	4 2 1
8	F8	F8	8	8
9	F9	F9	9	8 1
ƀ (Blank)	40	40	No Punch	— ③
. (period)	4B	4B	12, 8, 3	B A 8 2 1
: (colon)	7A	7A	8, 2	A ④
; (semicolon)	5E	5E	11, 8, 6	B 8 4 2
, (comma)	6B	6B	0, 8, 3	A 8 2 1
?	6F	6F	0, 8, 7	A 8 4 2 1
!	5A	5A	11, 8, 2	—
" (quote)	7F	7F	8, 7	8 4 2 1
' '(Prime)	} 7D	} 7D	} 8, 5	} 8 4 1
@	7C	7C	8, 4	8 4
¢	4A	4A	12, 8, 2	—

STANDARD CHARACTER CODES

CHARACTER ①	HEXADECIMAL		80-COLUMN CARD	7-TRACK TAPE (BCDIC)②
	CONSOLE KEYBOARD (EBCDIC)	EBCDIC		
$	5B	5B	11, 8, 3	B 8 2 1
%	6C	6C	0, 8, 4	A 8 4
&	50	50	12	B A
#	7B	7B	8, 3	8 2 1
*	5C	5C	11, 8, 4	B 8 4
– (underscore)	6D	6D	0, 8, 5	A 8 4 1
(	4D	4D	12, 8, 5	B A 8 4 1
)	5D	5D	11, 8, 5	B 8 4 1
\	—	6A	12, 11	—
/	61	61	0, 1	A 1
+	4E	4E	12, 8, 6	B A 8 4 2
– (minus)	60	60	11	B
=	7E	7E	8, 6	8 4 2
>	6E	6E	0, 8, 6	A 8 4 2
<	4C	4C	12, 8, 4	B A 8 4
	—	C0	12,0	B A 8 2
	—	D0	11,0	B 8 2
	—	E0	0, 8, 2	A 8 2
¬ (Not)	5F	5F	11, 8, 7	B 8 4 2 1
|	4F	4F	12, 8, 7	B A 8 4 2 1
SP	40		—	—
LF	25	—	—	—
CR	0D	—	—	—
ⓈEOM	37	—	—	—

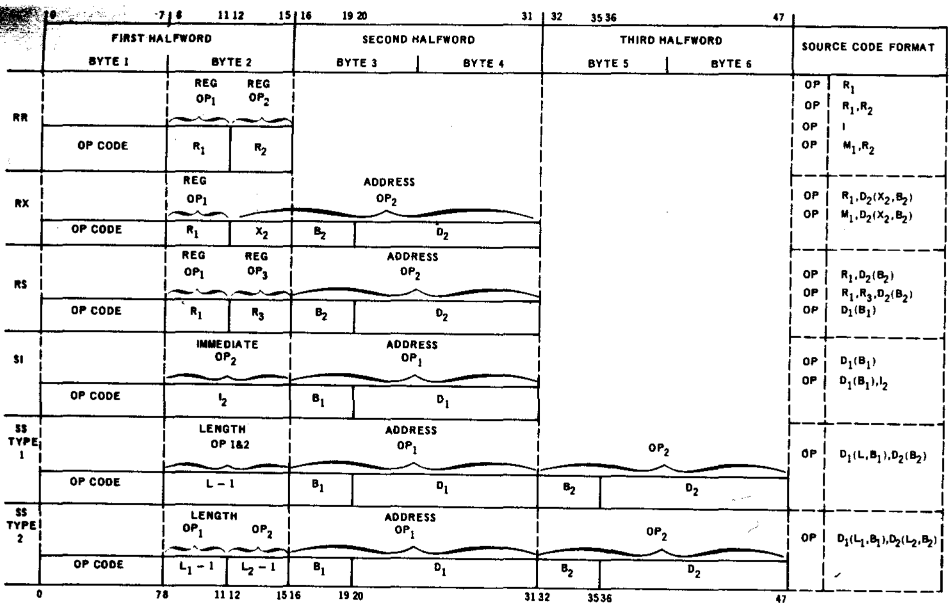
NOTES:

① Optional graphics are available.

② Hardware translate option is available on UNISERVO 12 and 16 Tape Units.

③ In odd parity, a blank (or space) corresponds to no pulse; in even parity, a blank corresponds to an A pulse.

④ In odd parity, reading an A pulse produces hexadecimal 7A (EBCDIC); in even parity, reading an A pulse produces hexadecimal 40.



SYMBOL	MEANING
OP CODE	THE INSTRUCTION OPERATION CODE
R_1	THE NUMBER OF THE GENERAL REGISTER WHICH HOLDS OPERAND 1
R_2	THE NUMBER OF THE GENERAL REGISTER WHICH HOLDS OPERAND 2
R_3	THE NUMBER OF THE GENERAL REGISTER WHICH HOLDS OPERAND 3
X_2	THE NUMBER OF THE GENERAL REGISTER WHICH HOLDS AN INDEX FOR OPERAND 2 OF AN RX INSTRUCTION
I	THE IMMEDIATE OPERAND
I_2	THE IMMEDIATE DATA OR DEVICE ADDRESS USED AS OPERAND 2 OF A SI INSTRUCTION
L	THE LENGTH SPECIFICATION OF THE FIRST OPERAND* (MAX. 256 BYTES)
L_1	THE LENGTH SPECIFICATION OF OPERAND 1* } (MAX. 16 BYTES)
L_2	THE LENGTH SPECIFICATION OF OPERAND 2* } (MAX. 16 BYTES)
B_1	THE NUMBER OF THE GENERAL REGISTER WHICH HOLDS THE BASE ADDRESS OF OPERAND 1
B_2	THE NUMBER OF THE GENERAL REGISTER WHICH HOLDS THE BASE ADDRESS OF OPERAND 2
D_1	THE DISPLACEMENT FOR OPERAND 1
D_2	THE DISPLACEMENT FOR OPERAND 2
OP_1	OPERAND 1
OP_2	OPERAND 2
OP_3	OPERAND 3
M_1	MASK

* *L represents the length as used in Source Code; not L – 1, as used in Object Code.*

A – ARITHMETIC INSTRUCTION SET

MNEMONIC	HEX. CODE	LOGIC DESCRIPTION	CONDITION* CODE GROUP
A	5A	$(R_1) + (D_2\ (X_2,B_2)) \rightarrow R_1$	ARITHMETIC
AH	4A	$(R_1) + \text{Sign expanded } (D_2\ (X_2,B_2)) \rightarrow R_1$	↓
AI	93	$I_2 + (D_1(B_1)) \rightarrow D_1(B_1)$ Halfword	↓
AP	FA	$(D_2(L_2,B_2)) + (D_1(L_1,B_1)) \rightarrow D_1(L_1,B_1); L_1 \geq L_2$	↓
AR	1A	$(R_1) + (R_2) \rightarrow R_1$	↓
DP	FD	† $(D_1(L_1,B_1)) \div (D_2(L_2,B_2)) \xrightarrow{QR} D_1(L_1,B_1); L_1 > L_2$	–
MP	FC	‡ $(D_1(L_1,B_1)) \times (D_2(L_2,B_2)) \rightarrow D_1(L_1,B_1); L_1 > L_2$	–
PACK	F2	$(D_2(L_2,B_2)) \xrightarrow{PACK} D_1(L_1,B_1)$	–
S	5B	$(R_1) - (D_2(X_2,B_2)) \rightarrow R_1$	ARITHMETIC
SH	4B	$(R_1) - \text{Sign expanded } (D_2(X_2,B_2)) \rightarrow R_1$	↓
SP	FB	$(D_1(L_1,B_1)) - (D_2(L_2,B_2)) \rightarrow D_1(L_1,B_1); L_1 \geq L_2$	↓
SR	1B	$(R_1) - (R_2) \rightarrow R_1$	↓
UNPK	F3	$(D_2(L_2,B_2)) \xrightarrow{UNPACK} D_1(L_1,B_1)$	–
ZAP	F8	$0 \rightarrow D_1(L_1,B_1); (D_2(L_2,B_2)) \rightarrow D_1(L_1,B_1); L_1 \geq L_2$	ARITHMETIC

*Refer to pages 12 and 13 for the condition code

L – LOGICAL INSTRUCTION SET

MNEMONIC	HEX. CODE	LOGIC DESCRIPTION	CONDITION* CODE GROUP
ED	DE	Expands $(D_2(L,B_2)) \rightarrow (D_1(L,B_1))$ per mask $D_1(L,B_1)$	GENERAL – a.
N	54	(R_1) AND $(D_2(X_2,B_2)) \rightarrow R_1$	LOGICAL
NC	D4	$(D_1(L,B_1))$ AND $(D_2(B_2)) \rightarrow D_1(L,B_1)$	↓
NI	94	$(D_1(B_1))$ AND $I_2 \rightarrow D_1(B_1)$	↓
NR	14	(R_1) AND $(R_2) \rightarrow R_1$	↓
O	56	(R_1) OR $(D_2(X_2,B_2)) \rightarrow R_1$	↓
OC	D6	$(D_1(L,B_1))$ OR $(D_2(B_2)) \rightarrow D_1(L,B_1)$	↓
OI	96	$(D_1(B_1))$ OR $I_2 \rightarrow D_1(B_1)$	↓
OR	16	(R_1) OR $(R_2) \rightarrow R_1$	↓
TM	91	$(D_1(B_1))$ Tested for 1 Bits as per I_2	GENERAL – d.
TR	DC	$(D_1(L,B_1)) \xrightarrow[\text{Look Up}]{\text{Table}} D_2(B_2) \rightarrow D_1(L,B_1)$	–
X	57	(R_1) XOR $(D_2(X_2,B_2)) \rightarrow R_1$	LOGICAL
XC	D7	$(D_1(L,B_1)$ XOR $(D_2(B_2)) \rightarrow D_1(L,B_1)$	↓
XI	97	$(D_1(B_1))$ XOR $I_2 \rightarrow D_1(B_1)$	↓
XR	17	(R_1) XOR $(R_2) \rightarrow R_1$	↓

† $D_1(L_1,B_1)$ must have at least one leading zero.
Quotient = Upper $L_1 - L_2$ bytes of $D_1(L_1,B_1)$
Remainder = Lower L_2 bytes of $D_1(L_1,B_1)$

‡ Lower $L_1 - L_2$ bytes of $D_1(L_1,B_1)$ = multiplier
Upper L_2 bytes of $D_1(L_1,B_1)$ are initially zero.

C – COMPARE INSTRUCTION SET

MNEMONIC	HEX. CODE	LOGIC DESCRIPTION	CONDITION * CODE GROUP
C	59	$(R_1): (D_2(X_2,B_2))$	COMPARE
CH	49	(R_1): Sign expanded $(D_2(X_2,B_2))$	↓
CL	55	$(R_1): (D_2(X_2,\ B_2))$	↓
CLC	D5	$(D_1(L,B_1)): (D_2(B_2))$	↓
CLI	95	$(D_1(B_1)): I_2$	↓
CLR	15	$(R_1): (R_2)$	↓
CP	F9	$(D_1(L_1,B_1)): (D_2(L_2,B_2)); L_1 \geq L_2$	↓
CR	19	$(R_1): (R_2)$	↓

B – BRANCH INSTRUCTION SET

MNEMONIC	HEX. CODE	LOGIC DESCRIPTION	CONDITION * CODE GROUP
BAL	45	$(PSW_{32\text{-}63}) \rightarrow R_1;\ D_2(X_2,B_2) \rightarrow PSW_{47\text{-}63}$	–
BALR	05	$(PSW_{32\text{-}63}) \rightarrow R_1;\ (R_2) \rightarrow PSW_{47\text{-}63}$	–
BC	47	M_1: cc; If M_1: cc, $D_2(X_2,B_2) \rightarrow PSW_{47\text{-}63}$	–
BCR	07	M_1: cc; If (M_1): $(R_2) \rightarrow PSW_{47\text{-}63}$	–
BCT	46	$(R_1) - 1 \rightarrow R_1$; If $(R_1) \neq 0) \rightarrow D_2(X_2,B_2) \rightarrow PSW_{47\text{-}63}$	–
BCTR	06	$(R_1) - 1 \rightarrow R_1$; If $(R_1) \neq 0\ (R_2) \rightarrow PSW_{47\text{-}63}$	–

D – DATA TRANSFER INSTRUCTION SET

MNEMONIC	HEX. CODE	LOGIC DESCRIPTION	CONDITION * CODE GROUP
L	58	$(D_2(X_2,B_2)) \rightarrow R_1$	–
LH	48	Sign expanded $(D_2(X_2,B_2)) \rightarrow R_{1\ 0\text{-}15}$	–
LM	98	$(D_2(B_2)) \rightarrow R_1 \ldots R_3$; Circular	–
LR	18	$(R_2) \rightarrow R_1$; set condition code	–
LTR	12	$(R_2) \rightarrow R_1$; set condition code	GENERAL – a.
MVC	D2	$(D_2(B_2)) \xrightarrow{CHAR} D_1(L,B_1)$; High order byte to low order byte	–
MVI	92	$I_2 \rightarrow D_1(B_1)$	–
MVN	D1	$(D_2(B_2)) \xrightarrow{NUM} D_1(L,B_1)$; High order byte to low order byte	–
MVO	F1	$(D_2(L_2,B_2)) \xrightarrow[\text{BITS LEFT}]{\text{OFFSET FOUR}} D_1(L_1,B_1)$	–
MVZ	D3	$(D_2(B_2)) \xrightarrow{\text{ZONE BITS}} D_1(L,B_1)$	–
SLL	89	$LS(R_1) \rightarrow R_1$ Shift K times. K = the contents of the lower order 6 bits of $D_2(B_2)$.	–
SRL	88	$RS(R_1) \rightarrow R_1$ End off and zero fill.	–
ST	50	$(R_1) \rightarrow D_2(X_2,B_2)$	–
STH	40	$(R_1)_{16\text{-}31} \rightarrow D_2(X_2,B_2)$ Halfword	–
STM	90	$(R_1),(R_1+1),\ldots,(R_3) \rightarrow D_2(B_2)$	–

*Refer to pages 12 and 13 for the condition code

E – EXTENDED MNEMONICS INSTRUCTION SET

MNEMONIC	HEX. CODE	M_1	SOURCE CODE FORMAT	CONDITION* CODE GROUP
FOR GENERAL USE				
B	47	F	$D_2(X_2,B_2)$	–
BR	07	F	R_1	–
NOP	47	0	$D_2(X_2,B_2)$	–
NOPR	07	0	R_1	–
AFTER COMPARISON OPERATIONS				
BE	47	8	$D_2(X_2,B_2)$	–
BER	07	8	R_1	–
BH	47	2	$D_2(X_2,B_2)$	–
BHR	07	2	R_1	–
BL	47	4	$D_2(X_2,B_2)$	–
BLR	07	4	R_1	–
BNE	47	7	$D_2(X_2,B_2)$	–
BNER	07	7	R_1	–
BNH	47	D	$D_2(X_2,B_2)$	–
BNHR	07	D	R_1	–
BNL	47	B	$D_2(X_2,B_2)$	–
BNLR	07	B	R_1	–

MNEMONIC	HEX. CODE	M_1	SOURCE CODE FORMAT	CONDITION* CODE GROUP
AFTER LOGICAL OPERATIONS				
BNZ	47	7	$D_2(X_2,B_2)$	–
BNZR	07	7	R_1	–
BZ	47	8	$D_2(X_2,B_2)$	–
BZR	07	8	R_1	–
AFTER TEST MASK OPERATIONS				
BM	47	4	$D_2(X_2,B_2)$	–
BMR	07	4	R_1	–
BNM	47	B	$D_2(X_2,B_2)$	–
BNMR	07	B	R_1	–
BNO	47	E	$D_2(X_2,B_2)$	–
BNOR	07	E	R_1	–
BO	47	1	$D_2(X_2,B_2)$	–
BOR	07	1	R_1	–

*Refer to pages 12 and 13 for the condition code

E – EXTENDED MNEMONICS INSTRUCTION SET (Cont.)

MNEMONICS	HEX. CODE	M_1	SOURCE CODE FORMAT	CONDITION* CODE GROUP
AFTER ARITHMETIC OPERATIONS				
BM	47	4	$D_2(X_2,B_2)$	–
BMR	07	4	R_1	–
BNM	47	B	$D_2(X_2,B_2)$	–
BNMR	07	B	R_1	–
BNO	47	E	$D_2(X_2,B_2)$	–
BNOR	07	E	R_1	–
BNP	47	D	$D_2(X_2,B_2)$	–
BNPR	07	D	R_1	–
BNZ	47	7	$D_2(X_2,B_2)$	–
BNZR	07	7	R_1	–
BO	47	1	$D_2(X_2,B_2)$	–
BOR	07	1	R_1	–
BP	47	2	$D_2(X_2,B_2)$	–
BPR	07	2	R_1	–
BZ	47	8	$D_2(X_2,B_2)$	–
BZR	07	8	R_1	–

*Refer to pages 12 and 13 for the condition code

P – PRIVILEGED INSTRUCTION SET

MNEMONICS	HEX. CODE	LOGIC DESCRIPTION	CONDITION* CODE GROUP
HPR	99	$D_1(B_1) \rightarrow$ DISPLAY IN THE A REGISTER: Halt CP without loss	–
LLR	81	$(D_2(B_2))_{0-7} \rightarrow$ Upper LR; $(D_2(B_2))_{8-15} \rightarrow$ Lower LR	GENERAL – b.
LPSW	82	$(D_1(B_1)) \rightarrow$ PSW register	–
SIO	9C	Initiate I/O Operation. $D_1+(B_1)$ specifies device address.	GENERAL – c.
SLM	B8	$(D_2(B_2)) \rightarrow R_1, R_1+1, \dots, R_3$	–
SSM	80	$(D_1(B_1))$ byte $\rightarrow$ PSW (System Mask)$_{0-6}$	–
SSTM	B0	$(R_1), (R_1+1), \dots, (R_3) \rightarrow D_2(B_2)$	–

S – SPECIAL INSTRUCTION SET

MNEMONICS	HEX. CODE	LOGIC DESCRIPTION	CONDITION* CODE GROUP
IC	43	$(D_2(X_2,B_2)) \rightarrow R_{1_{24-31}}$, $R_{1_{0-23}}$ Unchanged	–
LA†	41	$D_2(X_2,B_2) \rightarrow R_{1_{15-31}}$; $0 \rightarrow R_{1_{0-14}}$	–
SPM	04	$(R_1)_{2-5} \rightarrow PSW_{34-37}$	–
STC	42	$(R_1)_{24-31} \rightarrow D_2(X_2,B_2)$	–
SVC	0A	$1 \rightarrow$ Old PSW_{24-31}; $0 \rightarrow$ Old PSW_{16-23} Current $PSW_{0-15,32-63} \rightarrow$ Old $PSW_{0-15,32-63}$ New PSW $\rightarrow$ Current PSW	–

†For systems greater than 131 K bytes of memory, 18-bit addresses are generated.

	HEX. CODE	MNE-MONIC	TYPE	SET	TIMES (μ sec.)
RR	04	SPM	RR	S	6.0
	05	BALR	RR	B	6.0
	06	BCTR	RR	B	7.2
	07	BCR	RR	B	4.2
	0A	SVC	RR	S	7.8
	12	LTR	RR	D	4.8
	14	NR	RR	L	6.0
	15	CLR	RR	C	6.0
	16	OR	RR	L	6.0
	17	XR	RR	L	6.0
	18	LR	RR	D	4.8
	19	CR	RR	C	6.0
	1A	AR	RR	A	6.0
	1B	SR	RR	A	6.0

	HEX. CODE	MNE-MONIC	TYPE	SET	TIMES (μ sec.)
RX ①	40	STH	RX	D	4.2
	41	LA	RX	S	4.8
	42	STC	RX	S	4.2
	43	IC	RX	S	4.2
	45	BAL	RX	B	4.8
	46	BCT	RX	B	6.0
	47	BC	RX	B	3.0
	48	LH	RX	D	6.0
	49	CH	RX	C	6.0
	4A	AH	RX	A	6.0
	4B	SH	RX	A	6.0
	50	ST	RX	D	6.0
	54	N	RX	L	6.0
	55	CL	RX	C	6.0
	56	O	RX	L	6.0
	57	X	RX	L	6.0
	58	L	RX	D	4.8
	59	C	RX	C	6.0
	5A	A	RX	A	6.0
	5B	S	RX	A	6.0

	HEX. CODE	MNE-MONIC	TYPE	SET	TIMES (μ sec.)
RS & SI	80	SSM	SI	P	6.0
	81	LLR	RS	P	6.0
	82	LPSW	SI	P	7.2
	88	SRL	RS	D	See Note ④
	89	SLL	RS	D	$11.4+1.2C_1$
	90	STM	RS	D	$2.4+(2.4\times f)$
	91	TM	SI	L	6.0
	92	MVI	SI	D	4.2
	93	AI	SI	A	4.2
	94	NI	SI	L	4.2
	95	CLI	SI	C	4.2
	96	OI	SI	L	4.2
	97	XI	SI	L	4.2
	98	LM	RS	D	$2.4+(2.4\times f)$
	99	HPR	SI	P	6.0
	9C	SIO	SI	P	6.0 +CU Time ②
	B0	SSTM	RS	P	$2.4+(2.4\times f)$
	B8	SLM	RS	P	$2.4+(2.4\times f)$

	HEX. CODE	MNE-MONIC	TYPE	SET	TIMES (μ sec.)
SS	D1	MVN	SS1	D	$15.0+(2.4\times n)$
	D2	MVC	SS1	D	$15.0+(2.4\times n)$
	D3	MVZ	SS1	D	$15.0+(2.4\times n)$
	D4	NC	SS1	L	$15.0+(2.4\times n)$
	D5	CLC	SS1	C	$15.0+(2.4\times n)$
	D6	OC	SS1	L	$15.0+(2.4\times n)$
	D7	XC	SS1	L	$15.0+(2.4\times n)$
	DC	TR	SS1	L	$13.8+(4.8\times n_1)$
	DE	ED	SS1	L	$13.8+(3.6\times n)+(1.3\times n_s)$
	F1	MVO	SS2	D	$15.0+(2.4\times n_1)$
	F2	PACK	SS2	A	$12.6+(4.8\times n_1)$
	F3	UNPK	SS2	A	$15.0+(2.4\times n_1)$
	F8	ZAP	SS2	A	$15.0+(2.4\times n_1)$ ③
	F9	CP	SS2	C	$15.0+(2.4\times n_1)$ ③
	FA	AP	SS2	A	$15.0+(2.4\times n_1)$ ③
	FB	SP	SS2	A	$15.0+(2.4\times n_1)$ ③
	FC	MP	SS2	A	$21.6(n_1-n_2)$ ② $(n_2+2.68)-10.8n_2-15.3$
	FD	DP	SS2	A	$26.4(n_1-n_2)$ ② $(n_2+2.99)-10.8n_2-23.5$

① In the RX Instructions the timing is increased by 1.2 μsec when the index (x) field is not equal to zero.

② Yields an approximate time factor.

③ Plus $4.8+(2.4\times n_1)$ when the result may be complemented.

④ The timing of the SRL instruction is:

Condition	Time (μ sec)
$C_2 = 0$	11.4
$0 < C_2 < 16$	$54.0 - 2.4\ C_1$
$C_2 = 16$	15.6
$C_2 > 16$	$54.6 - 2.4\ C_1$
$C_2 = 32$ or 48	16.2

where:

C_1 = Least significant four bits of $D_2 + (B_2)$.

C_2 = Least significant six bits of $D_2 + (B_2)$

NOTES:

c	– the contents of the base register plus displacement.
f	– the number of full words.
n	– the number of result bytes.
n_1	– the number of bytes in operand one.
n_2	– the number of bytes in operand two.
n_s	– the number of signs in operand two.

ILLEGAL INSTRUCTIONS – 00, 2X, 3X, 6X, 7X, AX, BX, CX, EX, XF
where X = any value 0 through F

Illegal instructions decoded by the processor require 1.8 μs.

MASK OPERAND 1	DATA OPERAND 2	SIGNIFICANCE SWITCH SET ON	SIGNIFICANCE SWITCH SET OFF	SIGNIFICANCE SWITCH	BYTES REQUIRED IN MASK (OP1)
Fill Character	—	—	Remains in Mask	—	1 Byte
Digit Select Byte (DSB)	—	Inserts Zero or Digit	Inserts Fill Character or Digit	—	1 Byte for each Zero or Digit in Operand 2
Significant Start Byte (SSB)	—	Inserts Zero or Digit	Inserts Fill Character or Digit	Set to ON	
—	Zeros	Transferred to Operand 1 Mask	Replaced by Fill Character	—	Replaces DSB's or SSB's in the Mask Operand 1
—	Digits 1 through 9	Transferred to Operand 1 Mask	Transferred to Operand 1 Mask	Set to ON	
Comma	—	Remains in Mask Operand 1	Replaced by Fill Character	—	1 Byte for each Comma Used
Decimal Point	—	Remains in Mask Operand 1	Replaced by Fill Character	—	1 Byte for each Decimal Point
Other Characters	—	Remains in Mask Operand 1	Replaced by Fill Character	—	1 Byte for each Character Used
CR	—	Remains in Mask Operand 1	Replaced by Fill Character	—	2 Bytes if CR Used
– (Minus)	—	Remains in Mask Operand 1	Replaced by Fill Character	—	1 Byte if Minus Sign Used
—	POSITIVE SIGN ①	Not Transferred	Not Transferred	Set to OFF	—
—	NEGATIVE SIGN ①	Not Transferred	Not Transferred	Left unchanged	—
Field Separator Byte (FSB)	—	Replaced by Fill Character	Replaced by Fill Character	Set to OFF	1 Byte for each FSB Used

① The next high order position in Operand 2 is tested for a sign (value greater than nine) each time a digit is accessed and moved from Operand 2 to Operand 1. If positive (+), the S switch is turned off; if negative (–), the S switch is turned on. The sign is not moved to Operand 1. If the sign of Operand 2 is negative, Operand 1 data is retained. If the sign of Operand 2 is positive, Operand 1 data is deleted by fill characters.

CONDITION CODE SETTINGS

CONDITION CODE GROUP	INSTRUCTION	DECIMAL VALUE →	8	4	2	1
		BINARY CODE →	1000	0100	0010	0001
COMPARE	C CH CL CLC CLI CLR CP CR		=	<	>	NO CC
LOGICAL (AND/OR)	N NC NI NR O OC OI OR X XC XI XR		=0	≠0 (all 1's or mixed)	NO CC	NO CC

PROGRAM INTERRUPT CODES

HEX	CONDITION
0010	ILLEGAL OPERATION
0020	PRIVILEGED OPERATION
0030	– – –
0040	WRITE PROTECTION
0050	ADDRESSED EXCEPTION
0060	SPECIFICATION EXCEPTION
0070	WRITE PROTECT WITH ADDRESS EXCEPTION
0080	BINARY OVERFLOW*
0090	– – –
00A0	DECIMAL OVERFLOW*
00B0	DECIMAL DIVIDE EXCEPTION

*May be masked.

CONDITION CODE SETTINGS (cont.)

CONDITION CODE GROUP		INSTRUCTION	DECIMAL VALUE →	8	4	2	1
			BINARY CODE →	1000	0100	0010	0001
ARITHMETIC (ADD/ SUBTRACT)		A AH AI AP AR S SH SR SP		= 0	< 0	> 0	OVERFLOW
		ZAP					NO CC
GENERAL	a.	ED LTR		= 0	< 0	> 0	NO CC
	b.	LLR		WP=1	WP=0	NO CC	NO CC
	c.	SIO		I/O STARTED	CSW STORED	BUSY	NOT OP.
	d.	TM		= 0	MIXED	NO CC	= 1

NOTE: PSW Condition Code Settings

BITS 34, 35	00 – Test Value Is Binary 8 (1000)
	01 – Test Value Is Binary 4 (0100)
	10 – Test Value Is Binary 2 (0010)
	11 – Test Value Is Binary 1 (0001)

WP = Write Protection

CHANNEL COMMAND CODES

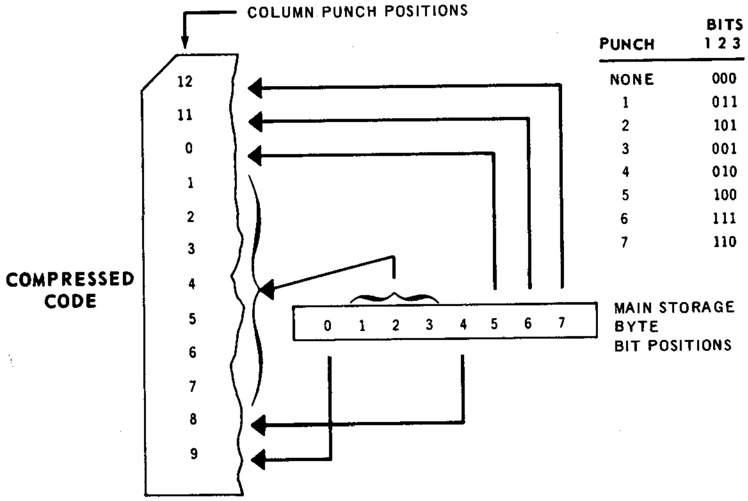
CODE*	COMMAND
0000 0000	TEST I/O
vvvv vv01	WRITE
vvvv vv10	READ
vvvv 0100	SENSE
0000 1000	TRANSFER IN CHANNEL
vvvv 1100	READ BACKWARDS
vvvv vv11	CONTROL

*v is a variable bit.

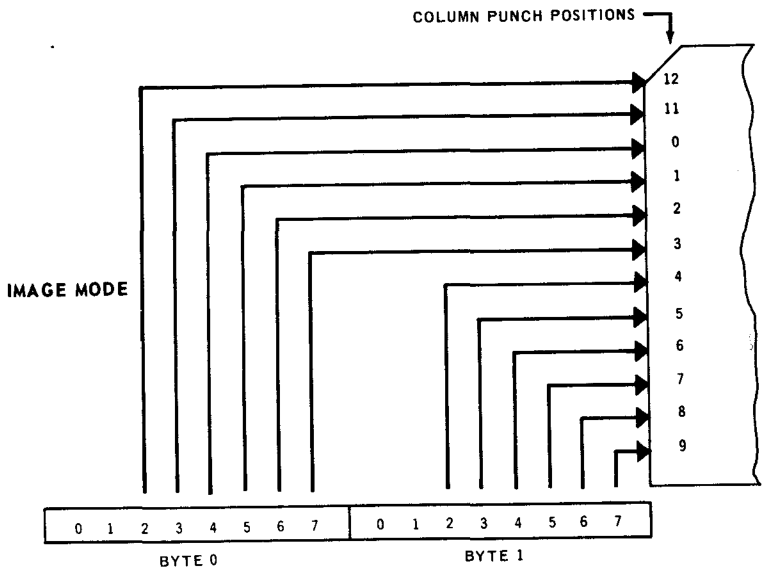
CARD CODE TRANSLATION (80-column cards)

PUNCH ONLY

PUNCH	BITS 1 2 3
NONE	000
1	011
2	101
3	001
4	010
5	100
6	111
7	110



NOTES: PUNCH POSITIONS 1 THROUGH 7 ARE INDICATED IN BITS 1 THROUGH 3 ACCORDING TO THE ABOVE TABLE. COMPRESSED CODE APPLICABLE TO THE CARD PUNCH ONLY; THE CARD READER TRANSLATES IT TO EBCDIC.



NOTE: BITS 0 AND 1 ARE CLEARED TO ZEROS ON AN IMAGE READ.

MULTIPLEXER CHANNEL

CHANNEL ADDRESS WORD (CAW)

IMMEDIATE STATUS STORAGE	0 0	COMMAND CODE
0 7	8 23	24 31

SHARED SUBCHANNEL CONTROL WORD (SCW)

DEVICE STATUS STORAGE	FLAG BITS T 0 0 0 0 D W	BUFFER CONTROL WORD INDEX
0 7	8 14	15 31

NONSHARED SUBCHANNEL CONTROL WORD

CONTROL BYTE	FLAG BITS T E P F C D W	BUFFER CONTROL WORD INDEX
0 7	8 14	15 31

SHARED BUFFER CONTROL WORD (BCW)

BYTE COUNT	0 0	0	DATA ADDRESS
0 11	12 13	14	15 31

NONSHARED BUFFER CONTROL WORD

BYTE COUNT	X X	0	DATA ADDRESS BCW ADDRESS
0 11	12 13	14	15 31

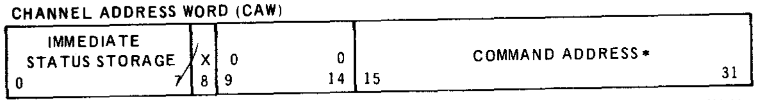
NOTE: BITS 12–13 00 STOP
01 LINK TO NON-CONTIGUOUS BCW
10 LINK TO NEXT BCW
11 STOP IMMEDIATE

STATUS WORD TABLE ENTRY (NONSHARED)

0 0	DEVICE NUMBER	DEVICE STATUS	SUBCHANNEL STATUS
0 7	8 15	16 23	24 31

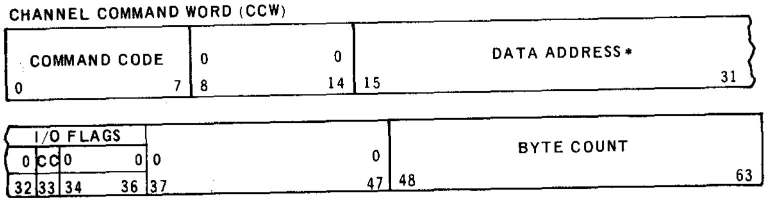
SELECTOR CHANNEL

CHANNEL ADDRESS WORD (CAW)

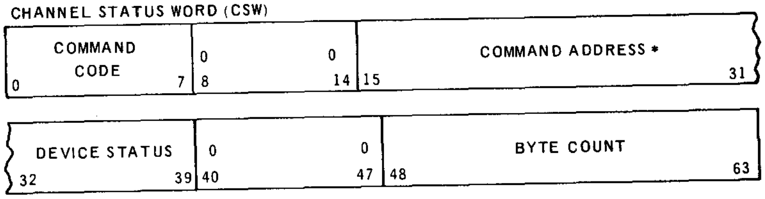


NOTE: BIT 8 USED FOR MAINTENANCE ONLY.

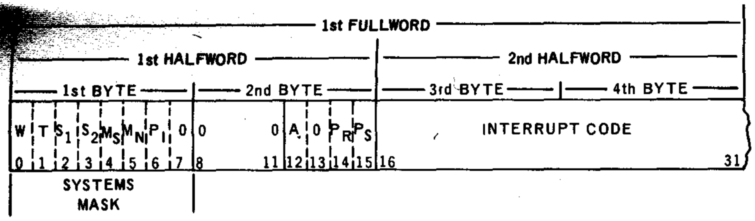
CHANNEL COMMAND WORD (CCW)

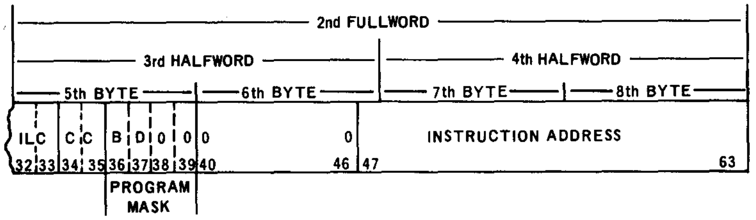


CHANNEL STATUS WORD (CSW)



*NOTE: FOR SYSTEMS GREATER THAN 131-K BYTES OF MAIN STORAGE, 18-BIT ADDRESSES (BITS 14–31) ARE GENERATED.

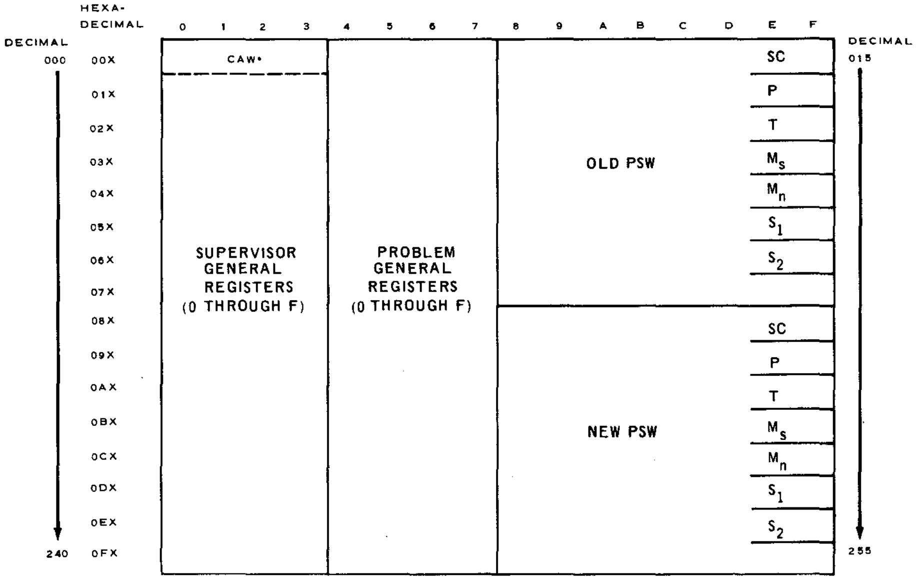


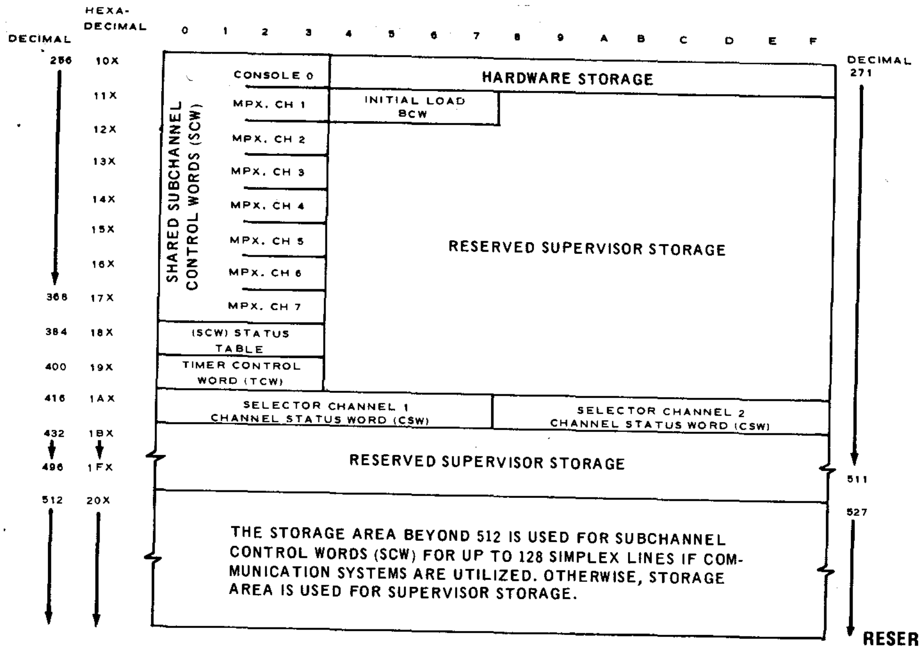


NOTE: FOR SYSTEMS GREATER THAN 131-K BYTES OF MAIN STORAGE, 18-BIT INSTRUCTION ADDRESSES (BITS 46–63) ARE GENERATED.

BIT(S)	ALLOCATION		FUNCTION
0	WRITE PROTECTION		1 – PERFORMED 0 – IGNORED
1 2 3 4 5 6	TIMER INTERRUPT SELECTOR CHANNEL 1 INTERRUPT SELECTOR CHANNEL 2 INTERRUPT MULTIPLEXER SHARED SUBCHANNEL INTERRUPT MULTIPLEXER NON-SHARED SUBCHANNEL INTERRUPT PROGRAM INTERRUPT		1 – ALLOWED 0 – INHIBITED
7–11	MUST BE 0		
12	INTERNAL OPERATING CODE		1 – ASCII 0 – EBCDIC
13	MUST BE 0		
14	REGISTER DESIGNATION		1 – PROBLEM GENERAL REGISTERS 0 – SUPERVISOR GENERAL REGISTERS
15	PROCESS STATE CONTROL		1 – PROBLEM STATE 0 – SUPERVISORY STATE
16–31	INTERRUPT CODE		SEE PROG. INTERRUPT CODES TABLE, PAGE 12
32,33	INSTRUCTION LENGTH CODE		00 – LENGTH NOT AVAILABLE 01 – 1 HALFWORD (RR) 10 – 2 HALFWORDS (RS, RX, SI) 11 – 3 HALFWORDS (SS1, SS2)
34,35	CONDITION CODE		00 – TEST VALUE IS BINARY 8 (1000) 01 – TEST VALUE IS BINARY 4 (0100) 10 – TEST VALUE IS BINARY 2 (0010) 11 – TEST VALUE IS BINARY 1 (0001)
36	PROGRAM MASK BITS	BINARY OVERFLOW EXCEPTION INTERRUPT	1 – ALLOWED 0 – INHIBITED
37		DECIMAL OVERFLOW EXCEPTION INTERRUPT	
38–46	MUST BE 0		
47–63	INSTRUCTION ADDRESS		(ADDRESS OF NEXT INSTRUCTION.)

RESERVED LOW ORDER MAIN STORAGE





THE STORAGE AREA BEYOND 512 IS USED FOR SUBCHANNEL CONTROL WORDS (SCW) FOR UP TO 128 SIMPLEX LINES IF COMMUNICATION SYSTEMS ARE UTILIZED. OTHERWISE, STORAGE AREA IS USED FOR SUPERVISOR STORAGE.

RESERVED LOW ORDER MAIN STORAGE (cont.)

CONVENTIONS

STATEMENT CONVENTIONS

CAPITAL LETTERS, PARENTHESES, AND PUNCTUATION MARKS	MUST BE CODED EXACTLY AS SHOWN
LOWERCASE LETTERS AND TERMS	REPRESENTS INFORMATION SUPPLIED BY THE PROGRAMMER
BRACES { }	NECESSARY ENTRIES OF WHICH ONE MUST BE CHOSEN
BRACKETS [ ]	OPTIONAL ENTRIES
ELLIPSIS . . .	INDEFINITE NUMBER OF ENTRIES

SUPERVISOR LINKAGE REGISTER CONVENTIONS

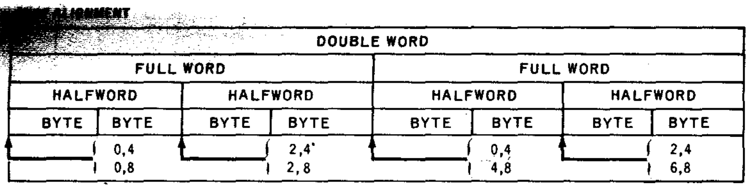
REGISTER	CONTENTS
0	PARAMETER REGISTER
1	PARAMETER/LIST REGISTER
2-12	FREE REGISTERS
13	SAVE AREA REGISTER
14	RETURN ADDRESS REGISTER
15	ENTRY POINT REGISTER

SIGN CONVENTIONS

HEXADECIMAL DIGIT	BINARY REPRESENTATION	SIGN VALUE	
A	1010	POSITIVE	*
B	1011	NEGATIVE	
C	1100	POSITIVE	EBCDIC
D	1101	NEGATIVE	
E	1110	POSITIVE	
F	1111	POSITIVE	

*Alternate sign convention

CNOP ALIGNMENT



DOUBLE WORD							
FULL WORD				FULL WORD			
HALFWORD		HALFWORD		HALFWORD		HALFWORD	
BYTE	BYTE	BYTE	BYTE	BYTE	BYTE	BYTE	BYTE
	0,4 0,8		2,4 2,8		0,4 4,8		2,4 6,8

ASSEMBLER AND DATA STORAGE STATEMENTS

LABEL	OPERATION	OPERAND	COMMENTS
[symbol]	CCW	Op_1,Op_2,Op_3,Op_4 (See note)	Defines and generates an eight-byte channel command word.

[symbol]	DC	[d] t [Ln] {'c'
		(c)}
			d = Duplication factor
			t = Type constant
			Ln = Length factor
			c = Constant specification

[symbol]	DS	[d] t [Ln] [{'c' (c)}]	

NOTE: OP_1 - COMMAND CODE　OP_3 - FLAGS
OP_2 - DATA ADDRESS　OP_4 - COUNT

ASSEMBLER ERROR FLAGS

LEVEL	MESSAGE	FLAG
FATAL	ESID OVERFLOW, MORE THAN 256 EXTERNALLY DEFINED SYMBOLS	B
	STORAGE OVERFLOW	F
	EXTRN OVERFLOW OF SYSPOOL AREA	K
	INTERNAL PROCESSING ERROR	V
DIAGNOSTIC	EXPRESSION NOT RELOCATABLE	A
	COVERING ERROR	C
	DUPLICATION	D
	EXPRESSION	E
	STATEMENT TOO LARGE	G
	OPERATION CODE ERROR	I
	LOCATION COUNTER OVERFLOW	L
	UNDEFINED SYMBOL	U
	CONTINUATION ERROR	X
	TOO MANY NESTED DO's OR PROC's	Z
ACADEMIC	CONDITIONAL ASSEMBLY ERROR	M
	NAME FIELD ERROR	N
	RELOCATION INFORMATION DROPPED	R
	STATEMENT OUT OF SEQUENCE	S
	TRUNCATION	T
	SYMBOL CONVERTED TO A CHARACTER STRING	W

DC/DS CONSTANT/STORAGE TYPES

t TYPE	ALIGNMENT	SPECIFIED BY	STORAGE FORMAT	TRUNCATION OR PADDING	LENGTH (BYTES) n		
					IMPLIED	MIN EXPLICIT	MAX EXPLICIT
C	NONE	CHARACTERS	CHARACTER	RIGHT	VARIABLE	1	DC-256 DS-65,536
X	NONE	HEXADECIMAL DIGITS	HEXADECIMAL	LEFT	VARIABLE	1	
B	NONE	BINARY DIGITS	BINARY	LEFT	VARIABLE	1	256
P	NONE	DECIMAL DIGITS	PACKED DECIMAL	LEFT	VARIABLE	1	16
Z	NONE	DECIMAL DIGITS	CHARACTER	LEFT	VARIABLE	1	16
H	HALFWORD	DECIMAL DIGITS	BINARY	LEFT	2	1	8
F	WORD	DECIMAL DIGITS	BINARY	LEFT	4	1	8
Y	HALFWORD	AN EXPRESSION	BINARY	LEFT	2	1	2
A	WORD	AN EXPRESSION	BINARY	LEFT	4	1	4
S	HALFWORD	ONE OR TWO ABSOLUTE EXPRESSIONS	BASE PLUS DISPLACEMENT	NONE	2	2	2
V	WORD	AN EXTERNAL SYMBOL	BINARY	LEFT	4	3	4

SUMMARY OF ASSEMBLER DIRECTIVES AND SUPERVISOR MACROS

ASSEMBLER

SYMBOL DEFINITION	EQU	LISTING CONTROL	EJECT PRINT SPACE TITLE
ASSEMBLER CONTROL	ASCII CNOP EBCDC END LTORG ORG START	INPUT/OUTPUT CONTROL	ICTL ISEQ PUNCH REPRO
BASE REGISTER ASSIGNMENT	DROP USING	CONDITIONAL ASSEMBLY	DO ENDO GBL GOTO LABEL LCL SET
PROGRAM LINKING AND SECTIONING	COM CSECT DSECT ENTRY EXTRN	SPECIAL DIRECTIVES	END NAME PNOTE PROC

SUPERVISOR

LABEL CONVENTIONS		STDEQU
PHYSICAL IOCS MACROS		CCB EXCP FREE MARK PIOCB RDFCB SWAP WAIT YIELD
PROGRAM MANAGEMENT MACROS	LOADING	FETCH LOAD
	TIMER CLOCK	GETIME SETIME
	TRANSIENT AREA	TCALL
	LINKAGE	CALL SAVE RETURN

USER LINKAGE	EXIT STXIT
SYSTEM INFO.	GETADR GETCOM PUTCOM
CONTROL STREAM	GETCS
TERMINATION	CANCEL DUMP EOJ SNAP
CHECKPT. & MESSAGES	CHKPT OPR
DISC ACCESS	GIVE QUERY TAKE
INSTRUCTION GENERATION	RELOC

JOB CONTROL

CONTROL STREAM STORAGE	// DELETE
	// FILE
			// RESET
			// RSTRT
			// SET
			// SKIP
			// VOL
			// n
			/$
			/*
			/&

JOB SELECTION	// RUN		

JOB SPECIFICATION	// ALTER
	// CANCEL
	// DVC
	// EQU
	// EXEC
	// EXT
	// FREE
	// JOB
	// LFD
	// LBL
	// MTC
	// OPR
	// OPTION
	// PARAM
		JOB CONTROL PROCEDURES (JPROC)	PROC
			NAME
			GOIF
			LABEL
			DATA
			REPL
			END
			&/
			&/n
			&$
			&*
			&&
			&b̶

OPERATOR COMMUNICATIONS

ALTER CANCEL DELETE DISPLAY DUMP FILE GO LIST LOG MOUNT MTC NOLOG PAUSE READY RUN SET CLOCK SET COMREG SET DATE SET IO SET SPSI STOP
DATA AND STORAGE STATEMENTS
CCW DC DS

COMMUNICATIONS CONTROL PROGRAM

ADVANCE BRANCH DATSTP DIRECT GETCP KYBUL LNEOPN LNEREL LNEREQ MPSTART MSGTYP PUTCP RECEND RECHDR RECPST RECSEG REROUTI REROUTO RETRANS ROUTE SENEND SENHDR SENPST SENSEG SEQIN SEQOUT SOURCE TIMSTP TRANSFER TRMCPY TRMREP UNICOR

SUMMARY OF JOB CONTROL, OPERATOR COMMUNICATIONS, AND CCP STATEMENTS

CLASS	OPERATOR	DESCRIPTION	HIERARCHY
ARITHMETIC OPERATOR	*/	A*/B IS EQUIVALENT TO $A*2^{B}$	6
	//	COVERED QUOTIENT, A//B IS EQUIVALENT TO (A + B – 1)/B	5
	/	A/B MEANS ARITHMETIC QUOTIENT OF A AND B	5
	*	A*B MEANS ARITHMETIC PRODUCT OF A AND B	5
	–	A–B MEANS ARITHMETIC DIFFERENCE OF A AND B	4
	+	A+B MEANS ARITHMETIC SUM OF A AND B	4
LOGICAL OPERATOR	**	A**B MEANS LOGICAL PRODUCT AND OF A AND B	3
	++	A++B MEANS LOGICAL SUM OR OF A AND B	2
	––	A––B MEANS LOGICAL DIFFERENCE XOR OF A AND B	2
RELATION OPERATOR	=	A=B HAS VALUE 1 IF TRUE HAS VALUE 0 IF FALSE	1
	>	A>B HAS VALUE 1 IF TRUE HAS VALUE 0 IF FALSE	1
	<	A<B HAS VALUE 1 IF TRUE HAS VALUE 0 IF FALSE	1

NOTE: The operations with the higher hierarchy numbers are performed first.
Operations are performed from left to right.

· ASSEMBLER OR SUPERVISOR	LABEL	OPERATION	OPERAND	DESCRIPTION
A		ASCII		DEFINES ASCII CONSTANT GENERATION FOLLOWING THE DIRECTIVE
S	[symbol]	CALL	{entry-point (15)} [,{(param-1,...,param-n) list-address (1)}]	USED TO ESTABLISH LINKAGE BETWEEN CALLING AND CALLED PROGRAM
S	[symbol]	CANCEL		USED TO TERMINATE CURRENT JOB STEP & ANY REMAINING STEPS IN A JOB
S	symbol	CCB	piocb-name, ccw-name[,entry-number] [,X'xx']	GENERATES A COMMAND CONTROL BLOCK FOR AN I/O DEVICE
S	[symbol]	CHKPT	{file-name,restart,file-list,error (1)}	CAUSES CHECK-POINT RECORDS TO BE WRITTEN

ASSEMBLER OR SUPERVISOR	LABEL	OPERATION	OPERAND	DESCRIPTION
S	[symbol]	CLSTA	loglist[,phylist][,csbuf]	GENERATES A CHECKPOINT FILE LIST
S	[symbol]	CLSTB	filename-1[,filename-2,...,filename-n]	GENERATES A LIST OF LOGICAL FILES FOR CHECKPOINT
S	[symbol]	CLSTC	(filename-1,tpmrk-count-name-1,blk-count-loc-1) [,(filename-2,tpmrk-count-name-2,blk-count-loc-2),...,(filename-n,tpmrk-count-name-n,blk-count-loc-n)]	GENERATES A LIST OF PHYSICAL FILES FOR CHECKPOINT
A		CNOP	a, a NOTE: param one – Location counter byte param two – Halfword, fullword, doubleword a is an absolute expression.	USED TO ADJUST THE LOCATION COUNTER TO HALFWORD, FULLWORD, OR DOUBLEWORD BOUNDARY
A		COM		DEFINES A CONTROL SECTION USED IN COMMON BY TWO OR MORE SEPARATELY ASSEMBLED ROUTINES
A	[symbol]	CSECT		INDICATES START OR CONTINUATION OF A CONTROL SECTION
A	SYSECT	CSECT		SYSECT CAUSES SUBSEQUENT CODE TO BE GENERATED IN THE CONTROL SECTION IN WHICH THE PROC IS CALLED
A	[symbol]	DO	b NOTE: b is a basic expression	GENERATES OBJECT OUTPUT CODE THE NUMBER OF TIMES SPECIFIED IN THE OPERAND
A		DROP	r1[,...]	INFORMS THE ASSEMBLER THAT REGISTERS ARE NO LONGER AVAILABLE
A	[symbol]	DSECT	NOTE: Label must be present in a program which has either no START statement, or a START or CSECT statement with a blank label field.	DEFINES DUMMY AREAS WITHIN A MODULE USED TO DESCRIBE AREAS DEFINED IN ANOTHER MODULE

ASSEMBLER OR SUPERVISOR	LABEL	OPERATION	OPERAND	DESCRIPTION
S	[symbol]	DUMP		GIVES A PRINTOUT OF MAIN STORAGE FOLLOWED BY TERMINATION OF THE JOB STEP
A		EBCDC		DEFINES EBCDIC CONSTANT GENERATION FOLLOWING THE DIRECTIVE
A		EJECT		CAUSES ADVANCEMENT TO THE NEXT PAGE OF LISTING
A	[symbol]	END	[e]	INDICATES THE END OF A SOURCE MODULE OR PROCEDURE DEFINITION BEING ASSEMBLED
A		ENDO		USED TO TERMINATE THE RANGE OF A DO STATEMENT
A		ENTRY	symbol [,symbol,...,symbol]	DECLARES SYMBOLS DEFINED WITHIN A MODULE
S	[symbol]	EOJ		NOTIFIES THE SUPERVISOR OF JOB STEP COMPLETION
A	symbol	EQU	e[,a] NOTE: e – Absolute or relocatable expression – value a – Absolute expression not exceeding a value of 256 – length	DEFINES EXPLICITLY THE VALUE AND LENGTH OF A SYMBOL
S	[symbol]	EXCP	$\left\{\begin{matrix}\text{ccb – name}\\(1)\end{matrix}\right\} \left[,\left\{\begin{matrix}C\\(0)\end{matrix}\right\}\right]$	SUBMITS I/O REQUESTS TO THE SYSTEM
S	[symbol]	EXIT	$\left[\left\{\begin{matrix}PC\\IT\\OC\end{matrix}\right\}\right]$	USED TO RETURN TO THE PROBLEM PROGRAM FROM THE USER'S INTERRUPT ROUTINE
A		EXTRN	symbol [,symbol,...,symbol]	SPECIFIES SYMBOLS USED IN A MODULE BUT DEFINED IN ANOTHER MODULE

ASSEMBLER OR SUPERVISOR	LABEL	OPERATION	OPERAND	DESCRIPTION
S	[symbol]	FETCH	{segment-name (1)} [,{entry-name (0)}]	LOCATES PROGRAM OVERLAY SEGMENTS IN THE EXECUTION AREAS, LOADS THEM IN MAIN STORAGE, AND TRANSFERS CONTROL TO THEM
S	[symbol]	FREE	{piocb-name (1)} [,{ALL entry-n (0)}]	RELEASES PERIPHERAL DEVICES FROM JOB STEP ASSIGNMENT
A		GBL	symbol [,symbol,...,symbol]	USED TO EXTERNALIZE OR DECLARE SET SYMBOLS WHICH ARE REFERENCED, DEFINED OR REDEFINED WITH PROCS
S	[symbol]	GETADR	[{SIB JCB PRE}] NOTE: Address is returned in register 1.	USED TO GET THE ABSOLUTE BASE ADDRESS OF THE SYSTEMS INFORMATION BLOCK, PROGRAMS JOB CONTROL BLOCK, AND THE PROGRAMS PREAMBLE
S	[symbol]	GETCOM	{to-address (r)}	RETRIEVES THE CONTENTS OF THE 12-BYTE COMMUNICATION REGION FROM THE PROGRAM PREAMBLE
S	[symbol]	GETCS	{input-area (1)} [,{# records (0)}]	READS A RECORD FROM THE JOB'S CONTROL STREAM
S	[symbol]	GETIME	[{S M}] NOTE: S – hhhmm M – binary milliseconds	USED TO GET THE TIME OF DAY FROM THE SIMULATED DAY CLOCK
S	[symbol]	GIVE	{list-name (1)}	USED TO REQUEST TEMPORARY STORAGE ASSIGNMENT
A		GOTO	symbol	USED TO ALTER THE SEQUENCE IN WHICH SOURCE STATEMENTS (OR PROCEDURE DEFINITION STATEMENTS) ARE PROCESSED BY THE ASSEMBLER

ASSEMBLER OR SUPERVISOR	LABEL	OPERATION	OPERAND	DESCRIPTION
A		ICTL	[b][,e][,c] NOTE: b – Begin column e – End column c – Continue column	SPECIFIES A NEW FORMAT FOR INPUT STATEMENTS
A		ISEQ	{ l,r | b̸ } NOTE: l – Leftmost column r – Rightmost column b̸ – end sequence checking	GIVES LOCATION OF SOURCE STATEMENT USED FOR SEQUENCE CHECKING AND INITIATES AND TERMINATES SEQUENCE CHECKING
A	symbol	LABEL		USED TO DEFINE LABELS USED BY THE GOTO STATEMENT
A		LCL	symbol [,symbol,...,symbol]	USED TO INITIALIZE SYMBOLS BEFORE THEY ARE REFERENCED OR DEFINED

S	[symbol]	LOAD	{ segment-name | (1) }
			{ segment-name | (1) } ,I
			{ address | (1) }
				LOCATES AND LOADS ABSOLUTE PROGRAM OVERLAY SEGMENTS INTO MAIN STORAGE
				LOCATES AN ENTRY WITHIN THE EXECUTION INDEX AND READS IT INTO A TEMPORARY WORK AREA IN THE PREAMBLE OF THE CALLING PROGRAM
				LOADS THE PROGRAM OVERLAY SEGMENTS RECORD SPECIFIED BY THE INDEX ENTRY IN THE PREAMBLE

A	[symbol]	LTORG		USED TO GENERATE ALL LITERALS PREVIOUSLY DEFINED
S	[symbol]	MARK	{ ccb-name | (1) } , { branch-addr | (15) }	CHECKS THE STATUS OF I/O OPERATIONS

ASSEMBLER OR SUPERVISOR	LABEL	OPERATION	OPERAND	DESCRIPTION
A	symbol	NAME	[p] NOTE: p is a parameter or a parameter sublist.	USED TO SPECIFY A NAME BY WHICH A PROC DEFINITION MAY BE CALLED (OR EVOKED)
S	[symbol]	OPR	$\left\{\begin{matrix}\text{message-address}\\(1)\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{length}\\(0)\end{matrix}\right\}\right]\left[,\left\{\begin{matrix}A\\I\\D\\R\end{matrix}\right\}\right]\left[,\left\{\begin{matrix}\text{REPLY}\\nn\end{matrix}\right\}\right]$ NOTE: A – Action I – Information D – Decision R – Ready command response	TRANSMITS MESSAGES TO THE SUPERVISOR FOR PRINTING AT THE CONSOLE
A	[symbol]	ORG	[e] NOTE: e is a relocatable expression.	USED TO SPECIFY, SET OR RESET A LOCATION COUNTER
S		OVRLAY	$\left\{\begin{matrix}\text{name}\\(1)\end{matrix}\right\},\left\{\begin{matrix}\text{svc-number}\\(0)\end{matrix}\right\}$	LOADS TRANSIENT OVERLAYS
S	[symbol]	PIOCB	$\left[\left\{\begin{matrix}\text{MAX}\\\text{\#-bytes}\end{matrix}\right\}\right]$ NOTE: #-bytes denotes the number of bytes in the file control blocks.	GENERATES A PHYSICAL I/O CONTROL BLOCK

ASSEMBLER OR SUPERVISOR	LABEL	OPERATION	OPERAND	DESCRIPTION
A	[symbol]	PROC	$\left[\left\{\begin{matrix}\text{symbol}\\\text{symbol, n}\\\text{symbol, n, k}\\\text{symbol, n, k, . . .}\end{matrix}\right\}\right]$ NOTE: n – number of positional parameters k – keyword parameters	USED TO SPECIFY THE BEGINNING OF PROC DEFINITION
A		PRINT	a, b, c NOTE: a – ON – Listing printed OFF – No listing b – GEN – PROC call generation NOGEN – No PROC object listing c – DATA – All characters printed NODATA – First eight characters printed	PROVIDES THE ABILITY TO CONTROL THE CONTENT OF THE LISTING – ONE, TWO OR THREE PARAMETERS CAN BE ENTERED IN ANY ORDER.
A		PNOTE	e,c NOTE: e-message character string c-comments character string	USED IN PROC DEFINITION TO GENERATE COMMENTS OR ERROR MESSAGES
S	[symbol]	PUTCOM	$\left\{\begin{matrix}\text{from-address}\\(1)\end{matrix}\right\}$	WRITES 12 BYTES OF INFORMATION IN THE COMMUNICATIONS REGION OF THE PROGRAM'S PREAMBLE

ASSEMBLER OR SUPERVISOR	LABEL	OPERATION	OPERAND	DESCRIPTION
S	[symbol]	QUERY	$\left\{\begin{matrix}\text{list-name}\\(1)\end{matrix}\right\}$	INTERROGATES THE USE OF BOTH ALLOCATED AND UNALLOCATED DIRECT ACCESS STORAGE
S	[symbol]	RDFCB	$\left\{\begin{matrix}\text{piocb-name}\\(1)\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{\#-bytes}\\(0)\end{matrix}\right\}\right]$	LOCATES A FILE CONTROL BLOCK IN THE JOB FILE AND READS IT INTO MAIN STORAGE
S		RELOC		CAUSES L Rx$,=A(ADDR-CONSTANTS)TO BE GENERATED IN PLACE OF LA INSTRUCTION
S	[symbol]	RETURN	[(r1,r2)][,T]	RELOADS REGISTERS SAVED BY THE SAVE MACRO AND RETURNS CONTROL
S	[symbol]	SAVE	[(r1,r2)][,T]	USED TO SAVE REGISTERS USED BY THE CALLED PROGRAM
A	symbol	SET	b NOTE: b is a basic expression.	USED TO DEFINE OR REDEFINE THE VALUE REPRESENTED BY A SYMBOL
S	[symbol]	SETIME	$\left[\left\{\begin{matrix}\text{time}\\(1)\end{matrix}\right\}\right]\left[\text{,WAIT}\right]$ NOTE: time in milliseconds	USED TO REQUEST INTERRUPTS BASED ON THE ELAPSING OF PROCESSING TIME

ASSEMBLER OR SUPERVISOR	LABEL	OPERATION	OPERAND	DESCRIPTION
S	[symbol]	SNAP	{ beginning-addr, ending-addr,...,addressing-pairs (1) }	USED TO DISPLAY THE CONTENTS OF THE 16 PROBLEM REGISTERS AND SELECTED MAIN STORAGE AREAS WITHIN THE PROBLEM PROGRAM
A		SPACE	[i] NOTE: i is an unsigned decimal integer	ADVANCES PAPER THE NUMBER OF LINES SPECIFIED
A	[symbol]	START	[a] NOTE: a is an absolute expression	DEFINES PROGRAM NAME AND TENTATIVE STARTING LOCATION
S	[symbol]	STDEQU	[{ G1 [HW][,SB][,JB][,JP][,IP][,DM][,MC] }]	USED TO GENERATE EQU DIRECTIVE FOR SYSTEM STANDARD LABELS

S	[symbol]	STXIT	(OC [, { entry-point,save-area,input-area, length (1) }]	USED TO ESTABLISH, CHANGE OR TERMINATE LINKAGE BETWEEN SUPERVISOR AND USER OPERATOR COMMUNICATION ROUTINE
			[{ IT PC }][, { entry-point,save-area (1),(0) }])	USED TO ESTABLISH SUPERVISOR LINKAGE WITH USERS PROGRAM CHECK AND INTERVAL TIMER ROUTINE
S	[symbol]	SWAP	{ piocb-name (1) } [, { entry-number (0) }]	SWAPS PHYSICAL UNIT BLOCK RELATIVE ADDRESSES
A			SYSNDX	USED TO CREATE UNIQUE NAMES FOR STATEMENTS WITHIN CODE GENERATED BY DIFFERENT CALL FOR THE SAME PROC

ASSEMBLER OR SUPERVISOR	LABEL	OPERATION	OPERAND	DESCRIPTION
S	[symbol]	TAKE	{ list-name (1) }	USED TO DEALLOCATE TEMPORARY DIRECT ACCESS STORAGE SPACE
S	[symbol]	TCALL	{ routine (1) } [, { param-1,...,param-n (0) }]	USED TO CALL TRANSIENT ROUTINES
A		TITLE	'c' NOTE: c is up to 100 characters of heading.	PROVIDES DATA FOR ASSEMBLER LISTING PAGE HEADINGS
S	[symbol]	TRLSE		TRANSIENT ROUTINE TERMINATION MACRO

A		USING	v,r1[,...]	USED FOR BASE REGISTER ASSIGNMENT AND DEFINITION OF SPECIFIC VALUES CONTAINED IN THE REGISTER
S	[symbol]	WAIT	{ ccb-name (1) ALL } [, { br-addr (15) }]	USED WHEN PROCESSING CANNOT PROCEED UNTIL I/O OPERATIONS ARE COMPLETE
S	[symbol]	YIELD		RELINQUISHES PROGRAM CONTROL

OPERATION	OPERAND	DESCRIPTION
// ALTER	$\left[\begin{Bmatrix}\text{PM}\\\text{RST}\\\text{A*address}\\\text{P*address}\\\text{R*address}\\\text{address}\\\text{ORG}\end{Bmatrix}\right]\left[,\begin{Bmatrix}\text{program-mark}\\\text{rst-address}\\\text{change}\\\text{org-address}\end{Bmatrix}\right]\left[,\begin{Bmatrix}\text{OPR}\\\text{RESET}\\\text{LAST}\end{Bmatrix}\right]$	USED TO INTRODUCE OBJECT CODE ALTERATIONS BY MEANS OF THE CONTROL STREAM AT PROGRAM RUN TIME
// CANCEL		USED TO TERMINATE THE PROCESSING OF THE JOB STREAM
// DELETE		USED TO DELETE A JOB STREAM FROM THE JOB FILE
// DVC	logical-unit $\left[,\begin{Bmatrix}\text{ALT}\\\text{a}\\\text{SYM}\\\text{ASYM}\end{Bmatrix}\right]$ [,STEP] [,uuu] [,OP] NOTE: a — any alphabetic letter (A through Z) that identifies a table listing the device type substitutions which can be used in assigning devices.	USED TO REQUEST THE ALLOCATION OF A PERIPHERAL DEVICE TO THE JOB
// EQU	lun-1, tt [,lun-2, tt,...,lun-5, tt]	USED TO EQUATE LOGICAL UNIT NUMBERS AND NAMES TO DEVICE CODES
// EXEC	program-name $\left[,\begin{Bmatrix}\text{library-name}\\\text{EX}\\\text{MCL}\end{Bmatrix}\right]$ [,filename] [,REL]	THE EXEC (EXECUTIVE) STATEMENT IS THE LAST STATEMENT PROCESSED BY THE CONTROL PROGRAM BEFORE PROBLEM PROGRAM EXECUTION
// EXT	$\left[\begin{Bmatrix}\text{C}\\\text{N}\end{Bmatrix}\right]$ [,ff] , $\begin{Bmatrix}\text{addr}\\\text{CYL}\\\text{TRK}\end{Bmatrix}$, quantity,. . . NOTE: C — Contiguous areas — cylinders or tracks must be assigned to contiguous areas N — Noncontiguous areas ff — ISAM Specifications 00 — not ISAM 01 — prime data area 02 — overflow data area 04 — Index See Note ①	USED TO SUPPLY INFORMATION FOR THE ESTABLISHMENT OF THE NEW OR THE EXTENSION OF EXISTING FILES ON DIRECT ACCESS DEVICES
// FILE	[n]	USED TO STORE CONTROL STREAMS ON THE RESIDENT DIRECT ACCESS DEVICE
// FREE	dev-1 [,dev-2,dev-3,...,dev-10]	USED TO DEALLOCATE A PERIPHERAL DEVICE FROM THE JOB
// JOB	job-name[,priority] [,time] [,min] [,max] [,nn] $\left[,\text{JOBLOG=}\begin{Bmatrix}\text{blank}\\\text{filename}\end{Bmatrix}\right]$	USED TO INDICATE THE BEGINNING OF CONTROL INFORMATION FOR A JOB

OPERATION	OPERAND	DESCRIPTION
// LBL	{file-id / 'file-id'} [,{file-serial-number / VCHECK}] [,volume-sequence-number] [,expiration-date] [,creation-date] [,file-sequence-number] [,generation-number] [,version-number]	USED TO SUPPLY LABEL INFORMATION FOR FILES ON DISC AND TAPE VOLUMES
// LFD	{filename / *filename} [,{SQ / DA / IS / DR}] [,n] [,{NEW / MISM}] [,ASC] *filename – input only file NOTE: SQ – Sequential DA – Direct Access IS – Index Sequential DR – Direct Access Relative b̸ – assumes SQ n – number of extents MISM – ignore printer mismatches for Data Management DTFPR files ASC – indicates an ASCII file See Note ①	RELATES THE PARTIAL DEFINITION OF A FILE IN A USER PROGRAM WITH THE COMPLETION OF THAT DEFINITION IN THE USER'S JOB CONTROL STREAM

① Continuation is allowed and is indicated with a nonblank in column 72.

OPERATION	OPERAND	DESCRIPTION
// MTC	logical-unit-number, {FM, nn / FB, nn / BM, nn / BB, nn / RL / RU / WM, nn}	USED TO POSITION TAPE VOLUMES PREVIOUSLY MOUNTED
// OPR	[*] comment-line NOTE: An asterisk before comment-line indicates a reply is necessary. The comment-line may extend through column 71.	USED TO DISPLAY A MESSAGE AT THE CONSOLE, REPLY WILL CAUSE A DELAY BETWEEN JOB STEPS
// OPTION	$p_1,...,p_8$ [ALTER, ALTER-NO, SYMBIONT, SYSDUMP, BOF, DOF, NOWP, NOVOL, NODUMP, MULTIFIL, NOREADY, RETRY, MAYIDUMP, {SCR, n} , {MCL, n}] NOTE: May be specified in any order or combination.	USED TO SPECIFY ONE OR MORE OF THE OPTIONS PROVIDED BY THE JOB CONTROL PROGRAM
// PARAM	ccc...c NOTE: c – character; length of character string may extend through column 71	SUBMITS PARAMETER TO A JOB STEP THROUGH THE JOB CONTROL STREAM
// RESET		SETS THE LOGICAL UNIT TABLE ASSOCIATED WITH A JOB EQUAL TO THE MASTER LOGICAL UNIT TABLE

OPERATION	OPERAND	DESCRIPTION
// RSTRT	logical-unit-number, serial-number[,filename]	USED TO RESTART A PROGRAM FROM A CHECKPOINT
// RUN	job-name[,priority] [,GO] [,partition-length] [,preamble-address]	CAUSES NORMAL TERMINATION OF THE CURRENT JOB AND INITIATION OF ANOTHER JOB
// SET	{ DATE,xx/xx/xx[,yyddd] [,yyddd] UPSI,switch-setting COMREG,character-string[,ASC] } NOTE: character string – X'ccccc' or C'ccccc'	USED TO SET UP OR MODIFY THE FOLLOWING PREAMBLE FIELDS: – DATE – SWITCH INDICATOR – COMMUNICATION REGION
// SKIP	[{ program-name n }] [,mask] NOTE: program-name – nnnnnnpp, format n – number of statements to be skipped	USED TO CONTROL THE SKIPPING OF CONTROL STATEMENTS BASED ON UPSI SETTING
// VOL	{ C Mcc ② CMcc ② vol. no.-1 SCRTCH } [, { vol. no.-1 or 2 SCRTCH }] [,vol. no.-2 or 3 through 7 or 8]	USED TO SUPPLY THE VOLUME SERIAL NO. FOR DATA AND PROGRAM VOLUMES TO BE ACCESSED BY THE JOB
// VOL	FORM,name,lines,lineno=h,lineno=e[,lineno=ss,...,lineno=ss]	USED TO DESCRIBE AND IDENTIFY A PAPER LOOP OR TO IDENTIFY A CARD TYPE TO BE USED BY THE COOP/ SYMBIONT
//n	$p_1,...,p_x$ NOTE: n – unsigned integer from 1 to 9.	USED TO INDICATE CONTINUATION
/$		INDICATES THE BEGINNING OF STREAMS OF DATA
/*		INDICATES THE END OF A DATA STREAM
/&		INDICATES THE END OF A CONTROL STREAM

② See page 48.

USED WITH UNISERVO VI-C SUBSYSTEMS

	cc	BYTES/INCH	PARITY	CONVERT FEATURE
SEVEN CHANNEL	10	200	ODD	ON
	20	200	EVEN	OFF
	30	200	ODD	OFF
	50	556	ODD	ON
	60	556	EVEN	OFF
	70	556	ODD	OFF
	90	800	ODD	ON
	A0	800	EVEN	OFF
	B0	800	ODD	OFF
NINE CHANNEL	80	800	ODD	OFF

USED TO SPECIFY CARD READER FEATURES REQUIRED – MODE PARAMETER ON VOL STATEMENT

READER TYPE	MODE SETTINGS	FEATURE
0716	04 08 10	1000 cpm Alternate stacker select or primary stacker full Dual translate feature (ASCII hardware translate)
0711/0716	20 40 80	66-column stub card feature 51-column stub card feature Validity check feature

CHARACTERS USED TO SPECIFY MODE SETTING ON VOL STATEMENT (pg. 46)

USED WITH UNISERVO 12 AND 16 SUBSYSTEMS

	cc	BYTES/INCH	PARITY	TRANSLATE FEATURE	CONVERT FEATURE
SEVEN CHANNEL	10	200	ODD	OFF	ON
	20	200	EVEN	OFF	OFF
	28	200	EVEN	ON	OFF
	30	200	ODD	OFF	OFF
	38	200	ODD	ON	QFF
	50	556	ODD	OFF	ON
	60	556	EVEN	OFF	OFF
	68	556	EVEN	ON	OFF
	70	556	ODD	OFF	OFF
	78	556	ODD	ON	OFF
	90	800	ODD	OFF	ON
	A0	800	EVEN	OFF	OFF
	A8	800	EVEN	ON	OFF
	B0	800	ODD	OFF	OFF
	B8	800	ODD	ON	OFF
NINE CHANNEL	C8	800	ODD	OFF	OFF
	C0	1600	ODD	OFF	OFF

// procname[.n] $\left[. \begin{Bmatrix} L \\ O \end{Bmatrix} \right]$ $[p_1,p_2,\ldots,p_n, k_1=v_1, k_2=v_2,\ldots,k_n=v_n]$

CALLS PREVIOUSLY ENTERED CONTROL STREAM PROCEDURES FROM PROCEDURE LIBRARY

NOTE:

The following six identifiers must be used on control statements and processor directives contained within procedures:

&/	–	Indicates Job Control statements within the proc body
&/n	–	Indicates continuation from previous card
&$	–	Indicates a Start-Of-Data statement within the proc body
&*	–	Indicates an End-Of-Data statement within the proc body
&&	–	Indicates an End-Of-Job statement within the proc body
&b̸	–	Indicates Librarian control statements within the proc body

Procedure Directives:

LABEL	OPERATION	OPERAND	DESCRIPTION
unused	PROC	$\begin{bmatrix} n,k \\ ,k \end{bmatrix}$	USED TO SPECIFY THE BEGINNING OF A PROC DEFINITION
symbol	NAME	unused	USED TO SPECIFY THE NAME BY WHICH A PROCEDURE IS REFERENCED
unused	END	unused	USED TO SPECIFY THE END OF A PROCEDURE
symbol	LABEL	unused	USED TO SPECIFY LABELS USED BY GOIF STATEMENTS
unused	GOIF	{symbol} [,b]	USED TO CONDITIONALLY OR UNCONDITIONALLY ALTER THE SEQUENCE IN WHICH STATEMENTS ARE GENERATED
unused	DATA	unused	USED TO INCLUDE DATA FROM THE INPUT READER IN A PROCEDURE GENERATED CONTROL STREAM
unused	REPL	unused	USED TO SPECIFY THAT PARAMETER PROCESSING BE PERFORMED ON DATA WITHIN THE PROC

ƀƀhh:mmƀCOMMANDƀPARAMETERS†	DESCRIPTION
ALTER [job-number,] [{PM RST A*address P*address R*address address ORG}] [,{program mask rst-address change org-address}] [,{RESET CARDS LAST}]	INTRODUCES OBJECT CODE ALTERATIONS BY MEANS OF SYSTEM CONSOLE AT RUN TIME
ANALYZE did	DISPLAYS INFORMATION ABOUT OUTSTANDING TAPE AND DISC ERRORS ON THE SYSTEMS LST DEVICE
CANCEL job-number, job-name[,NODUMP]	CAUSES CESSATION OF A JOB RUNNING IN THE SYSTEM
DELETE {job-name ALL}	CALLS THE DELETE FUNCTION OF THE JOB CONTROL PROGRAM
DISPLAY [job-number,] {PM RST A*address P*address R*address address} [,number bytes]	USED TO CAUSE THE PRINTING OF SELECTED AREAS OF MAIN STORAGE AT THE SYSTEM CONSOLE
DUMP {job-number, job-name SYSTEM}	PRINTS OUT MAIN STORAGE BY JOB OR SYSTEM
GO job-number	GO CHANGES THE JOB'S STATUS FROM NONREADY TO READY
FILE(nn) [,volume-serial-number [,{EOF m}]]	CALLS THE FILE FUNCTION OF THE JOB CONTROL PROGRAM
LIST {JOBS STOR JBFLE SIB job-number, job-name IO [,{DVC, did ALL TAPE DISC URCD type-code ERR}]}	CAUSES THE SYSTEMS INFORMATION BLOCK, JOB CONTROL BLOCK AND PROGRAM PREAMBLE, PHYSICAL UNIT BLOCKS, AND INDEX OF THE JOB FILE TO BE INCLUDED IN THE SYSTEMS OPERATION LOG
LOG	CAUSES ALL JOB CONTROL STATEMENTS TO BE INCLUDED IN THE SYSTEMS LOG
MOUNT pub-identifier [,volume-serial-number] [,s]	INFORMS THE SUPERVISOR THAT A VOLUME HAS BEEN MOUNTED ON A TAPE OR DISC PERIPHERAL DEVICE

†Pressing the ATTENTION key results in the printing of the time of day in the format hh:mm on the console printer. The command is typed next to the time by the operator. All commands can be specified by typing only the first two characters as underlined.

b̸b̸hh:mm b̸COMMAND b̸PARAMETERS†	DESCRIPTION
MTC pub-identifier, { FM,nn FB,nn BM,nn BB,nn RL RU WM,nn }	USED TO POSITION TAPE VOLUMES THAT HAVE BEEN MOUNTED ON TAPE UNITS
NOLOG	SUPPRESS LOGGING JOB CONTROL STATEMENTS EXCEPT JOB, EXEC, PAUSE, *, AND &
PAUSE jobnumber, user-comment	CAUSES A DELAY BETWEEN TWO JOB STEPS
READY job-number	INFORMS THE JOB CONTROL PROGRAM THAT REQUESTED OPERATOR ACTIONS HAVE BEEN COMPLETED
RUN job-name [,priority] [, GO] [,partition-length] [,preamble-address]	CALLS THE CONTROL PORTION OF THE JOB CONTROL PROGRAM
SET { CLOCK, hh:mm	USED TO SET THE SIMULATED CLOCK TO THE CORRECT TIME OF DAY
COMREG, character-string NOTE: character-string or 1–24 HEX ch X'xx...' 1–12 EBCDIC ch C'cc...'	USED TO STORE INFORMATION IN THE SYSTEMS COMMUNICATIONS REGION
DATE, xx/xx/xx[,yyddd] [,yyddd]	USED TO SET THE DATE FIELDS IN THE SYSTEM
IO, device-id, { DOWN UP SHARE NOSHARE CHANNEL, cochnl/chnl TYPE, type-code DEVICE, device-address VOLUME, volume-serial-number RES RDR IPT { LOG[,PCH] [,LST] PCH[,LST] [,LOG] LST[,LOG] [,PCH] } MODE, mode-setting COOP }	USED TO SET BITS IN THE PHYSICAL UNIT BLOCK AND MODIFY/SET SYSTEMS DEVICE ASSIGNMENTS
SPSI, switch-setting }	USED TO SET THE SYSTEM PROGRAM SWITCH
STOP jobnumber, jobname	SUSPENDS A JOB BETWEEN JOB STEPS

LABEL	OPERATION	OPERAND	DESCRIPTION
[symbol]	ADVANCE	{integer [, character-string] (1)}	MOVES MESSAGE HEADER SCAN POINTER OVER SPECIFIED NUMBER OF CHARACTERS OR OVER A CHARACTER STRING
[symbol]	BRANCH	branch-address	CAUSES BRANCH TO SPECIFIED ADDRESS WITHIN MPPS
[symbol]	DATSTP	[{N J}]	CAUSES DATE TO BE INSERTED INTO MESSAGE HEADER
[symbol]	DIRECT	{T,term-name P,processing-program-file ALTD SOURCE}	CAUSES MESSAGE TO BE PLACED ON A SPECIFIC DESTINATION QUEUE
[symbol]	GETCP	{file-name (1)} , {user-work-area-addr (0)}	CAUSES TRANSFER OF A MESSAGE FROM MCP TO UMPP
[symbol]	KYBUL		PLACES AN EOM AND KEYBOARD-UNDOCK CHARACTER AT END OF U300 MESSAGE
[symbol]	LNEOPN	line-name_1 [$,\text{line-name}_2$],...[$,\text{line-name}_n$]	INITIATES POLLING OF POLLABLE DEVICES AND INITIALIZES COMMUNICATIONS LINES
[symbol]	LNEREL	line-name	CAUSES COMMUNICATIONS LINES TO BE RELEASED FROM USER PROGRAM
[symbol]	LNEREQ	{line-name (1)}[,uoc, terminal-packet-addr, number of terminals]	REQUIRED TO HAVE LINE ASSIGNED TO USER PROGRAM
[symbol]	MPSTART	[reserved-byte-size]	REQUIRED DELIMITED FOR MPPS
[symbol]	MSGTYP	[integer, character-string, branch-addr]	USED TO SEPARATE INCOMING AND/OR OUTGOING MESSAGES INTO GROUPS
[symbol]	PUTCP	{file-name (1)} , {user-work-area-addr (0)}	TRANSFERS A MESSAGE FROM USER MPP TO MCP
[symbol]	RECEND		DELIMITER MACRO (REQ'D)
[symbol]	RECHDR		DELIMITER MACRO (REQ'D)
[symbol]	RECPST		DELIMITER MACRO (REQ'D)
[symbol]	RECSEG		DELIMITER MACRO (OPT'L)

COMMUNICATIONS CONTROL PROGRAM STATEMENTS (cont.)

LABEL	OPERATION	OPERAND	DESCRIPTION
[symbol]	REROUTI	mask, { SOURCE ALTD term-name }	REROUTE INPUT MESSAGE IF ANY ERROR BITS IN THE MASK MATCH THE ERROR HW
[symbol]	REROUTO	mask, { SOURCE ALTD term-name }	REROUTE OUTPUT MESSAGE IF ANY ERROR BITS IN THE MASK MATCH THE ERROR HW
[symbol]	RETRANS	mask,integer	ATTEMPT RETRANSMISSION INTEGER NUMBER OF TIMES IF ANY ERROR BITS IN THE MASK MATCH THE ERROR HW
[symbol]	ROUTE	eoa-character[,integer]	SCAN MESSAGE HEADER FOR ROUTING INFORMATION UNTIL EOA IS FOUND
[symbol]	SENEND		DELIMITER MACRO (REQ'D)
[symbol]	SENHDR		DELIMITER MACRO (REQ'D)
[symbol]	SENPST		DELIMITER MACRO (REQ'D)
[symbol]	SENSEG		DELIMITER MACRO (OPT'L)
[symbol]	SEQIN	[{ integer (1) }]	CAUSES COMPARISON TO BE MADE BETWEEN MESSAGE COUNTER IN MESSAGE HEADER AND TERMINAL
[symbol]	SEQOUT	{ integer (1) }	INSERTS A SEQUENCE NUMBER INTO OUTGOING MESSAGE
[symbol]	SOURCE	[{ integer (1) }]	SCAN MESSAGE HEADER TO VALIDATE SOURCE ID
[symbol]	TIMSTP		INSERT TIME IN MESSAGE HEADER
[symbol]	TRANSFER	[return-address]	BRANCH TO USER OWN CODE ADDRESS
[symbol]	TRMCPY	term-name ,work-area-addr [,no.-of-bytes]	TRANSFER TERMINAL CONTENTS FROM MCP TO USER AREA
[symbol]	TRMREP	term-name ,work-area-addr [,no.-of-bytes] (1)	TRANSFER TERMINAL CONTENTS FROM USER AREA TO TERMINAL TABLE

[symbol]	UNICOR	[{ U1 = (xx,xx)
		U3 = (xx,xx)
		U1 = (xx,xx), U3 = (xx,xx) }]
			INSERT COORDINATES IN MESSAGE DESTINED FOR A UNISCOPE 100 TERMINAL AND/OR UNISCOPE 300 TERMINAL

NOTE: x represents a hexadecimal digit.

CHARACTER CODES

DECIMAL	HEXA-DECIMAL	EBCDIC ①	80-COLUMN CARD CODE	CONSOLE KEYBOARD SET (EBCDIC)	7-TRACK TAPE (BCDIC)	COMPRESSED CARD CODE ②
0	00	NUL	12-0-9-8-1			NO PUNCH
1	01		12-9-1			3 ¦ 12
2	02		12-9-2			4 ¦ 11
3	03		12-9-3			1 ¦ 12,11
4	04	PF	12-9-4			5 ¦ 0
5	05	HT	12-9-5			2 ¦ 12,0
6	06	LC	12-9-6			7 ¦ 11,0
7	07	DEL	12-9-7			6 ¦ 12,11,0
8	08		12-9-8			9 ¦ 8
9	09		12-9-8-1			9,3 ¦ 8,12
10	0A		12-9-8-2			9,4 ¦ 8,11
11	0B		12-9-8-3			9,1 ¦ 8,12,11
12	0C		12-9-8-4			9,5 ¦ 8,0
13	0D		12-9-8-5	CARR. RET(CR)		9,2 ¦ 8,12,0
14	0E		12-9-8-6			9,7 ¦ 8,11,0
15	0F		12-9-8-7			9,6 ¦ 8,12,11,0
16	10		12-11-9-8-1			
17	11		11-9-1			
18	12		11-9-2			
19	13		11-9-3			
20	14	RES	11-9-4			
21	15	NL	11-9-5			
[illegible]	[illegible]	[illegible]	11-9-6			
23	17	IL	11-9-7			
24	18		11-9-8			
25	19		11-9-8-1			
26	1A		11-9-8-2			
27	1B		11-9-8-3			
28	1C		11-9-8-4			
29	1D		11-9-8-5			
30	1E		11-9-8-6			
31	1F		11-9-8-7			
32	20	DS	11-0-9-8-1			
33	21	SOS	0-9-1			
34	22	FS	0-9-2			
35	23		0-9-3			
36	24	BYP	0-9-4			
37	25	LF	0-9-5	LINE FEED(LF)		
38	26	EOB	0-9-6			
39	27	PRE	0-9-7			
40	28		0-9-8			
41	29		0-9-8-1			
42	2A	SM	0-9-8-2			
43	2B		0-9-8-3			
44	2C		0-9-8-4			
45	2D		0-9-8-5			
46	2E		0-9-8-6			
47	2F		0-9-8-7			
48	30		12-11-0-9-8-1			
49	31		9-1			

BIT POSITIONS 0, 1, 2, 3

BIT POSITIONS 4, 5, 6, 7

① Optional graphics are available.

② Punch patterns used to store the corresponding hexadecimal representation in the indicated bit positions of a byte.

DECIMAL	HEXA-DECIMAL	EBCDIC ①	80-COLUMN CARD CODE	CONSOLE KEYBOARD SET (EBCDIC)	7-TRACK TAPE (BCDIC)
50	32		9-2		
51	33		9-3		
52	34	PN	9-4		
53	35	RS	9-5		
54	36	UC	9-6		
55	37	EOT	9-7	Ⓢ (EOM)	
56	38		9-8		
57	39		9-8-1		
58	3A		9-8-2		
59	3B		9-8-3		
60	3C		9-8-4		
61	3D		9-8-5		
62	3E		9-8-6		
63	3F		9-8-7		
64	40	SP	NO PUNCHES	SPACE (SP)	
65	41		12-0-9-1		
66	42		12-0-9-2		
67	43		12-0-9-3		
68	44		12-0-9-4		
69	45		12-0-9-5		
70	46		12-0-9-6		
71	47		12-0-9-7		
72	48		12-0-9-8		
73	49		12-8-1		
74	4A	¢	12-8-2	¢	
75	4B	.	12-8-3	.	B A 8 2 1
76	4C	<	12-8-4	<	B A 8 4
77	4D	(	12-8-5	(	B A 8 4 1
78	4E	+	12-8-6	+	B A 8 4 2
79	4F	|(Vert. Bar)	12-8-7	|(Vert. Bar)	B A 8 4 2 1
80	50	&	12	&	B A
81	51		12-11-9-1		
82	52		12-11-9-2		
83	53		12-11-9-3		
84	54		12-11-9-4		
85	55		12-11-9-5		
86	56		12-11-9-6		
87	57		12-11-9-7		
88	58		12-11-9-8		
89	59		11-8-1		
90	5A	!	11-8-2	! or OR	
91	5B	$	11-8-3	$	B 8 2 1
92	5C	*	11-8-4	*	B 8 4
93	5D	)	11-8-5	)	B 8 4 1
94	5E	;	11-8-6	;	B 8 4 2
95	5F	¬ (Not)	11-8-7	¬(Not)	B 8 4 2 1
96	60	–	11	–	B
97	61	/	0-1	/	A 1
98	62		11-0-9-2		
99	63		11-0-9-3		
100	64		11-0-9-4		
101	65		11-0-9-5		
102	66		11-0-9-6		
103	67		11-0-9-7		
104	68		11-0-9-8		
105	69		0-8-1		
106	6A	\	12-11		
107	6B	,(Comma)	0-8-3	,(Comma)	A 8 2 1
108	6C	%	0-8-4	%	A 8 4

CHARACTER CODES (cont.)

DECIMAL	HEXA-DECIMAL	EBCDIC ①	80-COLUMN CARD CODE	CONSOLE KEYBOARD SET (EBCDIC)	7-TRACK TAPE (BCDIC)
109	6D	_(Underscore)	0-8-5	–(Underscore)	A 8 4 1
110	6E	>	0-8-6	>	A 8 4 2
111	6F	?	0-8-7	?	A 8 4 2 1
112	70		12-11-0		
113	71		12-11-0-9-1		
114	72		12-11-0-9-2		
115	73		12-11-0-9-3		
116	74		12-11-0-9-4		
117	75		12-11-0-9-5		
118	76		12-11-0-9-6		
119	77		12-11-0-9-7		
120	78		12-11-0-9-8		
121	79		8-1		
122	7A	:	8-2	:	A
123	7B	#	8-3	#	8 2 1
124	7C	@	8-4	@	8 4
125	7D	'(Prime or Apos)	8-5	'(Prime or Apos.)	8 4 1
126	7E	=	8-6	=	8 4 2
127	7F	" (Quotes)	8-7	" (Quotes)	8 4 2 1
128	80		12-0-8-1		
129	81	a	12-0-1		
130	82	b	12-0-2		
131	83	c	12-0-3		
132	84	d	12-0-4		
133	85	e	12-0-5		
134	86	f	12-0-6		
[illegible]	[illegible]	h	12-0-8		
137	89	i	12-0-9		
138	8A		12-0-8-2		
139	8B		12-0-8-3		
140	8C		12-0-8-4		
141	8D		12-0-8-5		
142	8E		12-0-8-6		
143	8F		12-0-8-7		
144	90		12-11-8-1		
145	91	j	12-11-1		
146	92	k	12-11-2		
147	93	l	12-11-3		
148	94	m	12-11-4		
149	95	n	12-11-5		
150	96	o	12-11-6		
151	97	p	12-11-7		
152	98	q	12-11-8		
153	99	r	12-11-9		
154	9A		12-11-8-2		
155	9B		12-11-8-3		
156	9C		12-11-8-4		
157	9D		12-11-8-5		
158	9E		12-11-8-6		
159	9F		12-11-8-7		
160	A0		11-0-8-1		
161	A1		11-0-1		
162	A2	s	11-0-2		
163	A3	t	11-0-3		
164	A4	u	11-0-4		
165	A5	v	11-0-5		
166	A6	w	11-0-6		
167	A7	x	11-0-7		
168	A8	y	11-0-8		
169	A9	z	11-0-9		
170	AA		11-0-8-2		
171	AB		11-0-8-3		
172	AC		11-0-8-4		
173	AD		11-0-8-5		
174	AE		11-0-8-6		
175	AF		11-0-8-7		
176	B0		12-11-0-8-1		
177	B1		12-11-0-1		
178	B2		12-11-0-2		
179	B3		12-11-0-3		
180	B4		12-11-0-4		
181	B5		12-11-0-5		
182	B6		12-11-0-6		
183	B7		12-11-0-7		
184	B8		12-11-0-8		
185	B9		12-11-0-9		
186	BA		12-11-0-8-2		
187	BB		12-11-0-8-3		
188	BC		12-11-0-8-4		
189	BD		12-11-0-8-5		
190	BE		12-11-0-8-6		
191	BF		12-11-0-8-7		
192	C0	PZ	12-0		B A 8 2
193	C1	A	12-1	A	B A 1
194	C2	B	12-2	B	B A 2
195	C3	C	12-3	C	B A 2 1
196	C4	D	12-4	D	B A 4
197	C5	E	12-5	E	B A 4 1
198	C6	F	12-6	F	B A 4 2
199	C7	G	12-7	G	B A 4 2 1
200	C8	H	12-8	H	B A 8
201	C9	I	12-9	I	B A 8 1
202	CA		12-0-9-8-2		
203	CB		12-0-9-8-3		
204	CC		12-0-9-8-4		
205	CD		12-0-9-8-5		
206	CE		12-0-9-8-6		
207	CF		12-0-9-8-7		
208	D0	MZ	11-0		B 8 2
209	D1	J	11-1	J	B 1
210	D2	K	11-2	K	B 2
211	D3	L	11-3	L	B 2 1
212	D4	M	11-4	M	B 4
213	D5	N	11-5	N	B 4 1
214	D6	O	11-6	O	B 4 2
215	D7	P	11-7	P	B 4 2 1
216	D8	Q	11-8	Q	B 8
217	D9	R	11-9	R	B 8 1
218	DA		12-11-9-8-2		
219	DB		12-11-9-8-3		
220	DC		12-11-9-8-4		
221	DD		12-11-9-8-5		
222	DE		12-11-9-8-6		
223	DF		12-11-9-8-7		
224	E0		0-8-2		A 8 2
225	E1		11-0-9-1		

① Optional graphics are available.

① Optional graphics are available.

DECIMAL	HEXA-DECIMAL	EBCDIC①	80-COLUMN CARD CODE	CONSOLE KEYBOARD SET (EBCDIC)	7-TRACK TAPE (BCDIC)
226	E2	S	0-2	S	A 2
227	E3	T	0-3	T	A 2 1
228	E4	U	0-4	U	A 4
229	E5	V	0-5	V	A 4 1
230	E6	W	0-6	W	A 4 2
231	E7	X	0-7	X	A 4 2 1
232	E8	Y	0-8	Y	A 8
233	E9	Z	0-9	Z	A 8 1
234	EA		11-0-9-8-2		
235	EB		11-0-9-8-3		
236	EC		11-0-9-8-4		
237	ED		11-0-9-8-5		
238	EE		11-0-9-8-6		
239	EF		11-0-9-8-7		
240	F0	0	0	0	8 2
241	F1	1	1	1	1
242	F2	2	2	2	2
243	F3	3	3	3	2 1
244	F4	4	4	4	4
245	F5	5	5	5	4 1
246	F6	6	6	6	4 2
247	F7	7	7	7	4 2 1
248	F8	8	8	8	8
249	F9	9	9	9	8 1
250	FA		12-11-0-9-8-2		
251	FB		12-11-0-9-8-3		
252	FC		12-11-0-9-8-4		
253	FD		12-11-0-9-8-5		
254	FE		12-11-0-9-8-6		
255	FF		12-11-0-9-8-7		

① Optional graphics are available.

[illegible] Then find the sum of the decimal values.

To convert to a hexadecimal, subtract the value in the table from the original decimal number that produces the smallest difference and note the hexadecimal equivalent. Repeat this procedure using the difference obtained in a previous calculation until a zero difference is obtained. The resulting hexadecimal representation (including zeros from columns not used) is the equivalent of the decimal number.

POWERS OF 16

16^n	n
1	0
16	1
256	2
4 096	3
65 536	4
1 048 576	5
16 777 216	6
268 435 456	7
4 294 967 296	8
68 719 476 736	9
1 099 511 627 776	10
17 592 186 044 416	11
281 474 976 710 656	12
4 503 599 627 370 496	13
72 057 594 037 927 936	14
1 152 921 504 606 846 976	15

POWERS OF 2

2^n	n
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16
131 072	17
262 144	18
524 288	19
1 048 576	20
2 097 152	21
4 194 304	22
8 388 608	23
16 777 216	24

BYTE				BYTE				BYTE			
0123		4567		0123		4567		0123		4567	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
6		5		4		3		2		1	

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
SEQUENTIAL	[name]	CLOSE	{filename filename-1,filename-2,...,filename-n (1)}	MUST BE USED TO TERMINATE ALL FILE PROCESSING
	[name]	CNTRL	{filename (1)} ,code [,m] [,n]	USED FOR PHYSICAL CONTROL OF SPECIFIED DEVICES
	[name]	FEOV	{filename (1)}	USED TO INITIATE END-OF-VOLUME PROCEDURES ON INPUT OR OUTPUT FILES
	[name]	GET	{filename (1)} [,{workarea (0)}]	USED TO OBTAIN ACCESS TO THE NEXT SEQUENTIAL LOGICAL INPUT RECORD
	[name]	LBRET	{1 2}	USED TO RETURN CONTROL FOLLOWING USER LABEL OR EXTENT PROCESSING
	[name]	OPEN	{filename filename-1,filename-2,...,filename-n (1)}	MUST BE USED TO INITIALIZE PROCESSING FOR ALL DATA MANAGEMENT FILES
			(1) [, (0)]	LOGICAL SEQUENTIAL OUTPUT RECORD
	[name]	RELSE	{filename (1)}	USED TO RELEASE THE CURRENT INPUT BLOCK
	[name]	TRUNC	{filename (1)}	PERMITS THE USER TO WRITE A SHORT BLOCK OF RECORDS
DIRECT	[name]	CLOSE	{filename filename-1,filename-2,...,filename-n (1)}	MUST BE USED TO TERMINATE ALL FILE PROCESSING
		CNTRL	{filename (1)}, SEEK	USED TO CONTROL OVERLAP BETWEEN SEEK ACTIVITY AND READ WRITE ACTIVITY ON NONSHARED UNITS
	[name]	LBRET	{1 2}	USED TO CREATE OR CHECK USER STANDARD HEADER LABELS AND TO PROCESS EXTENT INFORMATION
	[name]	OPEN	{filename filename-1,filename-2,...,filename-n (1)}	MUST BE USED TO INITIALIZE PROCESSING FOR ALL DATA MANAGEMENT FILES

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
DIRECT (cont.)	[name]	READ	{ filename } , { KEY } [,H] (1) ID	CAUSES A RECORD TO BE READ FROM THE DIRECT ACCESS STORAGE DEVICE INTO MAIN STORAGE
	[name]	RELEX	{ filename } [,ALL] (1)	USED TO REMOVE ENTRIES FROM THE LOCKOUT TABLE
	[name]	WAITF	{ filename } (1)	ENSURES THAT AN INITIATED COMMAND HAS BEEN COMPLETED, AND THAT ALL DATA HAS BEEN TRANSFERRED TO THE SPECIFIED AREA
	[name]	WRITE	{ filename } , { AFTER (1) AFTER, EOF RZERO }	USED TO WRITE A RECORD AFTER A SPECIFIED RECORD
	[name]	WRITE	{ filename } , { KEY } (1) ID	USED TO WRITE INTO A DEFINED AREA ON THE DISC

			...,filename-n	MINATE ALL FILE
	[name]	ENDFL	{ filename } (1)	CALLS ON A TRANSIENT ROUTINE TO TERMINATE FILE LOADING OR EXTENDING FUNCTIONS FOR THE FILE
	[name]	ESETL	{ filename } (1)	TERMINATES A RETRIEVAL SEQUENCE INITIATED BY SETL
	[name]	GET	{ filename } [, { workname }] (1) (0)	RETRIEVES THE NEXT LOGICAL RECORD IN SEQUENCE
	[name]	OPEN	{ filename filename-1,filename-2,...,filename-n (1) }	MUST BE USED TO INITIALIZE THE FILE BEFORE ANY OTHER IMPERATIVE MACRO INSTRUCTIONS CAN BE PERFORMED
	[name]	PUT	{ filename } [, { workname }] (1) (0)	INDICATES LAST RECORD RETRIEVED BY GET HAS BEEN UPDATED AND IS TO BE REWRITTEN ON THE DIRECT ACCESS STORAGE DEVICE

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
INDEXED SEQUENTIAL (cont.)	[name]	READ	{ filename (1) } , KEY	INITIATES THE RETRIEVAL OF A SINGLE LOGICAL RECORD FROM AN ISAM FILE
	[name]	SETFL	{ filename (1) }	CALLS ON A TRANSIENT ROUTINE WHICH SETS UP CONTROLS IN THE DTFIS MACRO, AND IN THE INDEXES ON THE DIRECT ACCESS STORAGE DEVICE, TO PREPARE THE FILE FOR LOADING (OR EXTENDING)
	[name]	SETL	{ filename (1) } , { idname (r) BOF GKEY KEY }	INITIALIZES A RETRIEVAL SEQUENCE; SPECIFIES THE FILE FROM WHICH RECORDS ARE TO BE RETRIEVED AND THE POINT AT WHICH RETRIEVAL IS TO START
	[name]	WAITF	{ filename (1) }	ENSURES THAT THE TRANSFER OF A RECORD BETWEEN MAIN STORAGE AND A DIRECT ACCESS STORAGE DEVICE HAS BEEN COMPLETED
	[name]	WRITE	{ filename (1) } , KEY	INITIATES THE REWRITING (UPDATING) OF THE LAST RECORD RETRIEVED WITH READ KEY
	[name]	WRITE	{ filename (1) } , NEWKEY	(1) CAN CAUSE A LOGICAL RECORD TO BE ADDED TO A FILE BEING LOADED OR EXTENDED (2) CAN INSERT A NEW RECORD INTO AN EXISTING FILE

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
SEQUENTIAL	filename	DTFCD	[AUE=YES]	ACCEPT MISPUNCHED CARDS FOR INPUT FILES (TYPE 0716 CARD READER ONLY)
			BKSZ=n	SPECIFIES MAXIMUM BLOCK SIZE IN BYTES
			[CNTRL=YES]	INDICATES THAT A CNTRL MACRO INSTRUCTION IS TO BE ISSUED FOR STACKER SELECTION ON THE ROW PUNCH SUBSYSTEM; CTLCHR MUST BE OMITTED
			[CTLCHR=YES]	USED WHEN A CONTROL CHARACTER IS TO BE USED IN DATA RECORDS; CNTRL MUST BE OMITTED
			EOFA=symbol	SPECIFIES THE ADDRESS TO WHICH CONTROL IS TRANSFERRED WHEN END OF DATA CARD FOR AN INPUT FILE IS SENSED
			[ERROR=symbol]	TRANSFERS CONTROL TO A SPECIAL HANDLING ROUTINE
			IOA1=symbol	RESERVES AN INPUT OR AN OUTPUT AREA FOR A FILE
			[IOA2=symbol]	RESERVES A SECOND INPUT OR A SECOND OUTPUT AREA FOR A FILE
			[IORG=(r),	SPECIFIED UNDER ANY OF THESE CONDITIONS: –WHEN A GENERAL REGISTER IS USED AS AN INDEX AREA TO REFERENCE CURRENT DATA –IF WORK AREA IS NOT REQUIRED AND THERE ARE TWO I/O AREAS –WHEN RECORDS ARE BLOCKED
			[ITBL=symbol]	USED WHEN RECORDS IN A GIVEN INPUT OR COMBINED FILE ARE TO BE TRANSLATED

			MODE= { BINARY
			STD
			CC
			TRANS }
				USED TO SPECIFY THE INPUT/OUTPUT MODE OF THE FILE

			[OBSZ=n]	USED FOR SPECIFYING THE LENGTH OF IOA2 FOR A COMBINED FILE
			[ORLP=YES]	USED WHEN THE ROW READ/PUNCH UNIT IS TO BE PROCESSED IN AN OVERLAP MODE
			[OPTION=YES]	USED TO SPECIFY AN OPTIONAL INPUT FILE

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
SEQUENTIAL (cont.)	filename (cont.)	DTFCD (cont.)	[OTBL=symbol]	USED TO TRANSLATE THE RECORDS IN AN OUTPUT OR COMBINED FILE
			[PUNR=YES]	USED TO ATTEMPT A CARD PUNCH ERROR RECOVERY

			RCFM= {FIXUNB
			UNDEF
			VARUNB}
				SPECIFIES FIXED-LENGTH UNBLOCKED RECORDS FOR INPUT AND COMBINED FILES, VARIABLE-LENGTH UNBLOCKED RECORDS FOR OUTPUT FILES, AND UNDEFINED RECORDS FOR OUTPUT FILES

			[RCSZ=(r)]	INDICATES THE NUMBER OF THE GENERAL REGISTER (2-12) THAT HOLDS THE SIZE OF THE OUTPUT RECORD

			STUB= {51
			66}
				SPECIFIES STUB CARD USAGE FOR TYPE 0716 CARD READER ONLY

			TYPE= {INPUT
			OUTPUT
			COMBND}
				DESCRIBES AN INPUT FILE FROM THE 600 CPM READER, AN OUTPUT FILE FOR THE ROW PUNCH SUBSYSTEM, AND THE COMBINED FILE OF THE ROW PUNCH SUBSYSTEM

			[WORK=YES]	SPECIFIED WHEN A WORK AREA, RATHER THAN THE I/O AREA, IS REQUIRED IN WHICH TO PROCESS I/O RECORDS
			BKSZ=n	SPECIFIES MAXIMUM BLOCK SIZE IN BYTES
			[CKPT=YES]	USED TO BYPASS CHECKPOINT RECORDS
			[CLRW= {RWD NORWD}]	USED TO REWIND WITHOUT INTERLOCK WHEN A FILE IS CLOSED, OR TO SPECIFY NO REWIND AT CLOSE
			EOFA=symbol	SPECIFIES ADDRESS TO WHICH CONTROL IS TRANSFERRED WHEN TAPE MARK FOLLOWING END OF DATA FOR AN INPUT FILE IS SENSED
			[ERRO= {IGNORE SKIP symbol}]	USED TO IGNORE THE ERROR AND PROCESS THE BLOCK, BYPASS THE BLOCK, OR ENTER A ROUTINE TO PROCESS THE ERROR BLOCK OR TERMINATE THE JOB
			[ERROR=symbol]	TRANSFERS CONTROL TO A SPECIAL ERROR HANDLING ROUTINE
			[FLBL= {STD NSTD NO}]	SPECIFIES STANDARD, NONSTANDARD, OR UNLABELED FILES
			IOA1=symbol	RESERVES AN INPUT OR AN OUTPUT AREA FOR A FILE

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
SEQUENTIAL (cont.)	filename (cont.)	DTFMT (cont.)	[IOA2=symbol]	RESERVES A SECOND INPUT OR A SECOND OUTPUT AREA FOR A FILE
			[IORG=(r)]	SPECIFIED UNDER ANY OF THESE CONDITIONS: –WHEN A GENERAL REGISTER IS USED AS AN INDEX AREA TO REFERENCE CURRENT DATA –IF WORK AREA IS NOT REQUIRED AND THERE ARE TWO I/O AREAS –WHEN RECORDS ARE BLOCKED
			[LBAD=symbol]	SUPPLIES THE ADDRESS OF A USER ROUTINE WHICH PROCESSES EITHER STANDARD (UHLn/UTLn) LABELS OR NONSTANDARD LABELS
			[OPTION=YES]	USED TO SPECIFY AN OPTIONAL INPUT FILE
			[OPRW=NORWD]	SPECIFIES NO REWIND BEFORE LABELS ARE CHECKED DURING PROCESSING OF THE OPEN MACRO INSTRUCTION
			[RCFM= {FIXUNB FIXBLK VARUNB VARBLK UNDEF}]	SPECIFIES FIXED-LENGTH UNBLOCKED OR BLOCKED, VARIABLE-LENGTH UNBLOCKED OR BLOCKED, OR UNDEFINED RECORDS

			READ= {FORWARD
			BACK}
				SPECIFIES TO LOGICAL IOCS THAT INPUT FILES ARE TO BE READ EITHER FORWARD OR BACKWARD

			[TPMK=NO]	ELIMINATES THE USE OF THE TAPE MARK
			[TYPE= {INPUT OUTPUT INOUT}]	SPECIFIES THE FILE OR FILES TO BE READ, WRITTEN, OR BOTH
			[VBLD=(r)]	FOR OUTPUT FILES WITH VARIABLE-LENGTH RECORDS TO BE PROCESSED IN THE I/O AREA, WHEN A WORK AREA IS NOT SPECIFIED
			[WORK=YES]	SPECIFIED TO PROCESS RECORDS IN WORK AREA
	filename	DTFOR	BKSZ=n	SPECIFIES MAXIMUM BLOCK SIZE IN BYTES
			[CNTRL=YES]	INDICATES THAT CNTRL MACRO WILL BE ISSUED TO THE FILE
			EOFA=symbol	SPECIFIES THE ADDRESS TO WHICH CONTROL IS TRANSFERRED WHEN END OF DATA IS SENSED

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
SEQUENTIAL (cont.)	filename (cont.)	DTFOR (cont.)	EOFB=n	SPECIFIES LENGTH OF END-OF-FILE SENTINEL
			EOFC=symbol	SPECIFIES THE ADDRESS TO WHICH CONTROL IS TRANSFERRED WHEN END-OF-FILE SENTINEL IS SENSED.
			[ERROR=symbol]	TRANSFERS CONTROL TO A SPECIAL ERROR HANDLING ROUTINE

			FEED= { 300
			 600 }
				SPECIFIES DOCUMENT FEED RATE OF 300 DPM OR 600 DPM

			IOA1=symbol	RESERVES AN INPUT AREA FOR A FILE
			[IOA2=symbol]	RESERVES A SECOND INPUT AREA FOR A FILE
			[IORG=(r)]	SPECIFIED UNDER THE FOLLOWING CONDITIONS: –WHEN TWO I/O AREAS ARE USED WITHOUT A WORK AREA –WHEN UNDEFINED RECORDS ARE PROCESSED WITHOUT A WORK AREA

			LGTH= n	SPECIFIES DOCUMENT LENGTH:
				n=3 FOR DOCUMENTS 3" – 3.2" IN LENGTH
				n=4 FOR DOCUMENTS 3.3" – 3.9" IN LENGTH
				n=5 FOR DOCUMENTS 4" – 5.9" IN LENGTH
				n=6 FOR DOCUMENTS 6" – 8.75" IN LENGTH

			[illegible]	[illegible] OF DATA READ BY OCR
			CARDB	SPECIFIES THAT PUNCHED CARD DATA IS TO BE READ IN IMAGE MODE
			CARDT	SPECIFIES THAT PUNCHED CARD DATA IS TO BE READ IN TRANSLATE MODE
			MARKB	SPECIFIES THAT MARK SENSE DATA IS TO BE READ IN IMAGE MODE
			MARKT	SPECIFIES THAT MARK SENSE DATA IS TO BE READ IN TRANSLATE MODE
			MODE= OCR	SPECIFIES THAT OCR DATA IS TO BE READ
			OCRCARDB	SPECIFIES THAT OCR WITH PUNCHED CARD DATA IS TO BE READ IN IMAGE MODE
			OCRCARDT	SPECIFIES THAT OCR WITH PUNCHED CARD DATA IS TO BE READ IN TRANSLATE MODE
			OCRMARKB	SPECIFIES THAT OCR WITH MARK SENSE DATA IS TO BE READ IN IMAGE MODE
			OCRMARKT	SPECIFIES THAT OCR WITH MARK SENSE DATA IS TO BE READ IN TRANSLATE MODE
			[OPTION=YES]	SPECIFIES AN OPTIONAL FILE

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
SEQUENTIAL (cont.)	filename (cont.)	DTFOR (cont.)	RCFM= $\begin{Bmatrix}\text{FIXUNB}\\\text{UNDEF}\end{Bmatrix}$	SPECIFIES THE RECORD FORMAT OF THE ODR FILE TO BE FIXED UNBLOCKED OR UNDEFINED
			[RCSZ=(r)]	SPECIFIES THE REGISTER (r)=2–12 THAT CONTAINS THE RECORD SIZE FOR UNDEFINED RECORDS
			ROWS= $\begin{Bmatrix}01\\23\\45\\67\end{Bmatrix}$	SPECIFIES MARK READ ROW SELECTION 01 – ROWS 0 & 1 23 – ROWS 2 & 3 45 – ROWS 4 & 5 67 – ROWS 6 & 7
			STKR= $\begin{Bmatrix}2\\3\end{Bmatrix}$	SPECIFIES OUTPUT STACKER 2 OR 3
			[STSL=symbol]	SPECIFIES THE ADDRESS OF THE STACKER SELECTION SUBROUTINE FOR SELECTING STACKERS ON AN INDIVIDUAL DOCUMENT BASIS
			[WORK=YES]	SPECIFIES THAT A WORK AREA, RATHER THAN THE I/O AREA, IS TO BE USED TO PROCESS RECORDS

				CONFIGURATION THAT IS NOT A PRINTABLE CHARACTER
			BKSZ=n	SPECIFIES MAXIMUM BLOCK SIZE IN BYTES
			[CNTRL=YES]	SPECIFIED IF SPACING OR SKIPPING OF LINES ON THE PRINTER IS CONTROLLED FROM THE PROBLEM PROGRAM BY THE CNTRL MACRO INSTRUCTION. CTLCHR MUST BE OMITTED
			[CODE=symbol]	SUPPLIES THE PRINTER CODE CONVERSION TABLE
			[CTLCHR=YES]	SPECIFIED WHEN A CONTROL CHARACTER IS TO BE USED IN DATA RECORDS. CNTRL MUST BE OMITTED
			[ERROR=symbol]	TRANSFERS CONTROL TO A SPECIAL ERROR HANDLING ROUTINE
			IOA1=symbol	RESERVES AN AREA OUTPUT FILE
			[IOA2=symbol]	RESERVES A SECOND INPUT OR A SECOND OUTPUT AREA FOR A FILE

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
SEQUENTIAL (cont.)	filename (cont.)	DTFPR (cont.)	[IORG=(r)]	SPECIFIED UNDER ANY OF THESE CONDITIONS: —WHEN A GENERAL REGISTER IS USED AS AN INDEX AREA TO REFERENCE CURRENT DATA —IF WORK AREA IS NOT REQUIRED AND THERE ARE TWO I/O AREAS
			[PRAD=n]	SPECIFIES FORM ADVANCE OF ONE, TWO, OR THREE LINES
			[PRTOV= { YES symbol }]	SPECIFIES THE OPERATION TO BE PERFORMED WHEN A CARRIAGE OVERFLOW CONDITION OCCURS, EITHER HOME PAPER OR USER'S OVERFLOW ROUTINE

			RCFM= { FIXUNB
			VARUNB
			UNDEF }
				SPECIFIES FIXED-LENGTH UNBLOCKED RECORDS FOR PRINT FILES, VARIABLE-LENGTH UNBLOCKED RECORDS, OR UNDEFINED RECORDS

			[RCSZ=(r)]	USED FOR PRINT FILE WITH UNDEFINED FORMAT
			[WORK=YES]	USED WHEN A WORK AREA, RATHER THAN I/O AREA, IS REQUIRED TO PROCESS OUTPUT RECORDS
			EOFA=symbol	SPECIFIES THE ADDRESS TO WHICH CONTROL IS TRANSFERRED WHEN END OF DATA IS SENSED
			[EOR=expression]	DEFINES END-OF-RECORD CHARACTER FOR OUTPUT FILES. REQUIRED FOR UNDEF OUTPUT FILES
			[ERRO= { IGNORE SKIP symbol }]	SPECIFIES TO IGNORE AN ERROR BLOCK, BYPASS THE ERROR BLOCK, ENTER A ROUTINE TO PROCESS THE ERROR BLOCK, OR TERMINATE THE JOB
			[ERROR=symbol]	SPECIFIES THAT CONTROL IS TO BE TRANSFERRED TO A SPECIAL ERROR HANDLING ROUTINE
			[FSCAN=symbol]	SPECIFIES THE ADDRESS OF THE FIGURE SCAN TABLE FOR OUTPUT FILES
			[FTRANS=symbol]	SPECIFIES THE ADDRESS OF THE FIGURE TRANSLATION TABLE FOR INPUT FILES
			IOA1=symbol	RESERVES AN INPUT OR AN OUTPUT AREA FOR THE FILE

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
SEQUENTIAL (cont.)	filename (cont.)	DTFPT (cont.)	[IOA2=symbol]	RESERVES A SECOND INPUT OR A SECOND OUTPUT AREA FOR THE FILE
			[IORG=(r)]	SPECIFIES THAT THE RECORDS ARE TO BE PROCESSED IN THE I/O AREA AND THAT THERE ARE TWO I/O AREAS
			[LSCAN=symbol]	SPECIFIES THE ADDRESS OF THE LETTER SCAN TABLE FOR OUTPUT FILES
			[LTRANS=symbol]	SPECIFIES THE ADDRESS OF THE LETTER TRANSLATION TABLE FOR INPUT FILES
			[MODE= {BINARY STD}]	SPECIFIES CHARACTER RECOGNITION OR PARITY GENERATION –BINARY – SUPPRESS CHARACTER RECOGNITION OR PARITY GENERATION –STD – CHARACTER RECOGNITION OR PARITY GENERATION EFFECTIVE
			[OBKS=n]	SPECIFIES MAXIMUM I/O AREA SIZE FOR SHIFTED CODE
			[OPTION=YES]	SPECIFIES AN OPTIONAL INPUT FILE
			[RCFM= {FIXUNB UNDEF}]	SPECIFIES FIXED-LENGTH UNBLOCKED OR UNDEFINED RECORDS
			[SCAN=symbol]	SPECIFIES THE ADDRESS OF THE SCAN TABLE FOR INPUT FILES
			[TRANS=symbol]	SPECIFIES THE ADDRESS OF THE TRANSLATION TABLE
			[TYPE= {INPUT OUTPUT}]	SPECIFIES TYPE OF FILE
			[WORK=YES]	SPECIFIES THAT RECORDS ARE TO BE PROCESSED IN A WORK AREA
	filename	DTFSD	BKSZ=n	SPECIFIES MAXIMUM BLOCK SIZE IN BYTES
			[CNTRL=YES]	INDICATES CNTRL MACRO WILL BE ISSUED TO THE FILE
			[DEVICE= {8411 8414}]	SPECIFIES THE TYPE OF DIRECT ACCESS STORAGE DEVICE UPON WHICH THE DATA FILE RESIDES
			[EOFA=symbol]	SPECIFIES ADDRESS TO WHICH CONTROL IS TRANSFERRED WHEN END OF DATA FOR AN INPUT FILE IS SENSED

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
SEQUENTIAL (cont.)	filename (cont.)	DTFSD (cont.)	[ERRO= {IGNORE SKIP symbol}]	USED TO IGNORE THE ERROR AND PROCESS THE BLOCK, BYPASS THE BLOCK, ENTER A SPECIAL ERROR PROCESSING ROUTINE, OR TERMINATE THE JOB
			[ERROR=symbol]	TRANSFERS CONTROL TO A SPECIAL ERROR HANDLING ROUTINE
			IOA1=symbol	RESERVES AN INPUT OR AN OUTPUT AREA FOR A FILE
			[IOA2=symbol]	RESERVES A SECOND INPUT OR A SECOND OUTPUT AREA FOR A FILE
			[IORG=(r)]	SPECIFIED UNDER ANY OF THESE CONDITIONS: –WHEN A GENERAL REGISTER IS USED AS AN INDEX AREA TO REFERENCE CURRENT DATA –IF WORK AREA IS NOT REQUIRED AND THERE ARE TWO I/O AREAS –WHEN RECORDS ARE BLOCKED
			[LBAD=symbol]	SUPPLIES THE ADDRESS OF THE ROUTINE WHICH PROCESSES USER STANDARD LABELS
			[OPTION=YES]	SPECIFIES AN OPTIONAL INPUT FILE
			[RCFM= {[illegible] VARUNB VARBLK UNDEF}]	[illegible] OR BLOCKED, VARIABLE-LENGTH UNBLOCKED OR BLOCKED, OR UNDEFINED
			[RCSZ= {n (r)}]	MAKES THE NUMBER OF BYTES IN A RECORD AVAILABLE TO LOGICAL IOCS
			[TRUNCS=YES]	APPLIES ONLY TO FIXED-LENGTH, BLOCKED RECORDS OF A FILE. IDENTIFIES AN INPUT FILE THAT CONTAINS SHORT BLOCKS
			[TYPE= {INPUT OUTPUT INOUT}]	SPECIFIES THE FILE THAT IS TO BE READ, WRITTEN, OR BOTH
			[UPDT=YES]	USED TO UPDATE AN INPUT FILE ON A DIRECT ACCESS STORAGE DEVICE
			[VBLD=(r)]	FOR PROCESSING OUTPUT FILES WITH VARIABLE-LENGTH RECORDS IN THE I/O AREA, WHEN A WORK AREA IS NOT SPECIFIED
			[VERIFY=YES]	FOR CHECK-READING RECORDS AFTER THEY HAVE BEEN WRITTEN

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
SEQUENTIAL (cont.)	filename (cont.)	**DTFSD (cont.)**	**[WORK=YES]**	FOR TRANSFERRING INPUT FILE RECORDS FROM THE INPUT I/O AREA TO A WORK AREA
DIRECT	filename	**DTFDA**	**[AFTER=YES]**	SPECIFIED IF A SUBSEQUENT WRITE MACRO INSTRUCTION CONTAINS AN AFTER OR RZERO POSITIONAL PARAMETER
			BKSZ=n	SPECIFIES MAXIMUM BLOCK SIZE IN BYTES
			[CNTRL=YES]	INDICATES CNTRL MACRO WILL BE ISSUED TO THE FILE
			$\left[\text{DEVICE=}\begin{Bmatrix}8411\\8414\end{Bmatrix}\right]$	DEFINES THE TYPE OF DEVICE UPON WHICH THE DATA FILE EXISTS
			ERRBYTE=symbol	SETS ERROR CONDITIONS OR STATUS CODE INTO TWO-BYTE FIELD PROVIDED BY THE USER
			[ERROR=symbol]	TRANSFERS CONTROL TO A SPECIAL ERROR HANDLING ROUTINE
			[illegible]	[illegible] ONE JOB
			[IDLOC=symbol]	STORES RECORD ID IN A FIVE-BYTE FIELD AFTER CERTAIN READ OR WRITE MACRO INSTRUCTIONS ARE EXECUTED
			IOA1=symbol	RESERVES AN INPUT OR AN OUTPUT AREA FOR A FILE
			[KEYARG=symbol]	SPECIFIES THAT RECORDS ARE TO BE IDENTIFIED BY A KEY
			[KEYLEN=n]	SPECIFIES THAT RECORDS ARE TO CONTAIN OR ARE TO BE REFERENCED BY KEYS
			[LBAD=symbol]	SUPPLIES ADDRESS OF USER ROUTINE TO PROCESS USER HEADER LABELS
			[READID=YES]	USED WHEN THE RECORD TO BE READ IS TO BE LOCATED BY ITS ADDRESS (ID)
			[READKEY=YES]	USED IF THE RECORD TO BE READ IS TO BE LOCATED BY ITS KEY

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION

DIRECT (cont.)	filename (cont.)	DTFDA (cont.)	RCFM= { FIXUNB
			UNDEF }
				SPECIFIES THAT RECORDS ARE EITHER FIXED-LENGTH UNBLOCKED, OR UNDEFINED IN LENGTH

			[RCSZ=(r)]	SPECIFIED IF THE DATA LENGTH OF RECORDS IS UNDEFINED
			[RELATIVE= { R T }]	SPECIFIES THE TYPE OF RELATIVE ADDRESSING TO BE USED BY THE PROBLEM PROGRAM
			SEEKADR=symbol	SPECIFIES THE ADDRESS OF AN EIGHT-BYTE FIELD IN THE PROBLEM PROGRAM
			[SRCHM=YES]	EXTENDS A SEARCH ON KEY TO MULTIPLE TRACKS

			TYPE= { INPUT
			OUTPUT }
				SPECIFIES THAT STANDARD LABELS ARE TO BE READ OR WRITTEN FOR THIS FILE

			[VERIFY=YES]	REQUESTS THAT A PARITY CHECK BE MADE OF ALL RECORDS AFTER THEY HAVE BEEN WRITTEN ON THE DEVICE
			[WRITEID=YES]	SPECIFIED IF AN OUTPUT RECORD IS TO BE LOCATED BY MEANS OF ITS ADDRESS (IR)
			[XTNTXIT=symbol]	CAUSES THE INFORMATION IN AN EXTENT TO BE PASSED TO THE USER
INDEXED SEQUENTIAL	filename	DTFIS	[CLOSE= { (NOWRITE[,DISPLAY]) (DISPLAY) }]	TWO UNRELATED ACTIONS. ALLOWS HARDWARE FILE PROTECT TO BE APPLIED AND/OR TO DISPLAY ON CONSOLE CONCATENATED FILENAME FIELDS
			[{ CYLOFL=n PCYLOFL=nn }]	SPECIFIES THE NUMBER OF TRACKS IN EACH PRIME DATA CYLINDER RESERVED FOR CYLINDER OVERFLOW FOR A FILE BEING LOADED; SPECIFIES THE NUMBER OF TRACKS RESERVED FOR CYLINDER OVERFLOW AS A PERCENTAGE OF THE NUMBER OF TRACKS PER CYLINDER
			[DEVICE= { 8411 8414 }]	DEFINES THE TYPE OF DEVICE WHICH CONTAINS THE DATA FILE
			[EOFA=symbol]	SPECIFIES ADDRESS TO WHICH CONTROL IS TRANSFERRED WHEN END OF DATA IS SENSED
			[ERROR=symbol]	TRANSFERS CONTROL TO A SPECIAL ERROR HANDLING ROUTINE

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
INDEXED SEQUENTIAL (cont.)	filename (cont.)	DTFIS (cont.)	[INDAREA=symbol]	SPECIFIES THE SYMBOLIC NAME OF THE MAIN STORAGE AREA ASSIGNED TO HOLD THE CYLINDER INDEX
			[INDSIZE=n]	SPECIFIES THE NUMBER OF BYTES AVAILABLE FOR CYLINDER INDEX

			IOAREAL=symbol [,IOAREAR=symbol][,IOAREAS=symbol]
			IOAREAR=symbol [,IOAREAS=symbol]
			IOAREAS=symbol
				USED TO SPECIFY LOADING OR ADDING TO THE FILE, RETRIEVING AND UPDATING IN RANDOM ORDER, OR RETRIEVING AND UPDATING IN SEQUENTIAL ORDER

			[IORG=(r)]	SPECIFIES THE GENERAL REGISTER TO BE USED TO STORE THE ADDRESS OF THE I/O AREA WHEN IT IS USED IN PLACE OF WORK AREAS
			IOROUT= ADD ADDRTR LOAD RELOAD RETRVE	ADD AND THE RETRIEVE FUNCTIONS ARE PERFORMED, THAT EITHER A NEW FILE IS CREATED OR AN OLD FILE IS EXTENDED, THAT AN EXISTING FILE IS BEING RECREATED IN THE SAME DISC SPACE, OR THAT RECORDS ARE RETRIEVED (AND/OR UPDATED) EITHER RANDOMLY OR SEQUENTIALLY
			[IOSIZE=n]	SPECIFIES A MULTIPLE OF THE MINIMUM NUMBER OF BYTES ALLOCATED TO THE IOAREAL AREA
			[KEYARG=symbol]	SPECIFIES THE ADDRESS (SYMBOL) IN THE PROBLEM PROGRAM OF A FIELD WHICH CONTAINS THE KEY TO BE USED AS A SEARCH ARGUMENT
			KEYLEN=n	SPECIFIES THE KEY LENGTH, IN BYTES, FOR THE FILE
			[KEYLOC=n]	SPECIFIES THE LOCATION OF THE KEY FIELD FOR ALL RECORDS OF THE FILE WHEN BLOCKED RECORDS ARE TO BE PROCESSED

ACCESS METHOD	LABEL	OPERATION	OPERAND	DESCRIPTION
INDEXED SEQUENTIAL (cont.)	filename (cont.)	DTFIS (cont.)	[MSTIND=YES]	SPECIFIED IF A MASTER INDEX IS TO BE CONSTRUCTED FOR A FILE BEING LOADED
			[NRECDS=n]	SPECIFIES THE NUMBER OF LOGICAL RECORDS PER PHYSICAL BLOCK IF BLOCKED RECORDS ARE TO BE PROCESSED

			RCFM= {FIXUNB
			FIXBLK}
				SPECIFIES EITHER BLOCKED OR UN-BLOCKED RECORDS IN AN ISAM FILE

			RCSZ=n	SPECIFIES FIXED LENGTH RECORDS ARE AN ISAM FILE
			[TYPE= {RANDOM SEQNTL RANSEQ}]	SPECIFIES IF RANDOM, SEQUENTIAL, OR RANDOM AND SEQUENTIAL PROCESSING IS TO BE PERFORMED FOR ISAM FILES ON WHICH RETRIEVAL FUNCTIONS ARE TO BE PERFORMED
			[VERIFY=YES]	REQUESTS THAT A PARITY CHECK BE MADE OF DATA RECORDS AFTER THEY HAVE BEEN WRITTEN ON A DEVICE
			[{[,WORKR=symbol] WORKL=symbol [,WORKR= symbol] WORKR=symbol}]	THE ADDRESS OF THE WORK AREA, IF RECORDS ARE TO BE RETRIEVED (AND UPDATED) RANDOMLY, AND WHERE THE SYMBOL IS THE ADDRESS OF THE WORK AREA; OR IF RECORDS ARE RETRIEVED (AND UPDATED) SEQUENTIALLY, AND WHERE EACH GET AND PUT IMPERATIVE MACRO INSTRUCTION WILL PROVIDE THE ADDRESS OF A WORK AREA

LABEL	OPERATION	OPERAND	DESCRIPTION
	ALLOC	{piocb-name-of-fcb (1)}, {piocb-name-of-erb (0)}	USED TO REQUEST THE SERVICES OF THE DISC SPACE MANAGEMENT ALLOCATE ROUTINE
	SCRTCH	{param-list (1)} [, {extent-number ALL (0)}]	USED TO REQUEST THE SERVICES OF THE DISC SPACE MANAGEMENT SCRATCH ROUTINE
	RENAME	{param-list (1)}	USED TO REQUEST THAT THE KEY FIELD OF A FORMAT 1 RECORD IN VTOC BE CHANGED
	OBTAIN	{param-list (1)}	USED TO REQUEST THAT A RECORD IN VTOC BE READ INTO A SPECIFIED AREA

TYPE	LABEL	OPERATION	OPERAND	DESCRIPTION
SORT FILE DEFINITION	pmt†	MR$PRM	ADTABL=symbol	USED TO SPECIFY AN ADDITIONAL, OR REFERENCE A PREVIOUSLY CREATED, PARAMETER TABLE
			AUTO=(label-code,sort-filename [,rec-per-cycle])	USED WHEN ALL PARTS OF A LARGE VOLUME SORT ARE TO BE EXECUTED AUTOMATICALLY

			BIN= { bytes
			(min-bytes,size-1,vol-1,..., size-n,vol-n) }
				REQUIRED FOR VARIABLE-LENGTH RECORDS; VARIABLE LENGTH RECORDS ARE DIVIDED INTO FIXED BIN SIZES TO CONSERVE STORAGE AND PROVIDE OPTIMUM SPEED IN PHASE 1

			CSPRAM= { YES
			NO
			OPTION }
				USED TO ACCESS CONTROL STREAM PARAMETERS, BYPASS THEM, OR ALLOW THE OPERATOR TO INDICATE THEIR PRESENCE AT RUN TIME

			DISC=max-disc-filenumber	INDICATES THAT DISC UNITS MAY BE USED BY THE SORT FOR SCRATCH PURPOSES
			[,rec-per-cycle][,number-merge-tape])	LARGE VOLUME, NON-AUTOMATIC SORT IS TO BE EXECUTED
			DOEXT=symbol	RETURNS CONTROL TO PROBLEM PROGRAM AT THE SPECIFIED ADDRESS, AT COMPLETION OF PART A OR M OF LARGE VOLUME SORT
			DOF=(in-label-1[,in-label-2,...,in-label-14])	USED WHEN PART F OF A LARGE VOLUME, NON-AUTOMATIC SORT IS TO BE EXECUTED
			DOM=(label-code,sort-filename,cycle number,in-label-1,in-label-2 [,...,in-label-13])	USED WHEN PART M OF A LARGE VOLUME, NON-AUTOMATIC SORT IS TO BE EXECUTED

			DROC= { symbol
			DELETE }
				SPECIFIED WHEN THE PROBLEM PROGRAM IS TO ELIMINATE OR COMBINE RECORDS WITH EQUAL KEY FIELDS

			FIELD=(byte-pos-1,lgth-1[,form-1][,seq-1] [,order-1],...,byte-pos-n, lgth-n[,form-n][,seq-n][,order-n])	REQUIRED IF THE SORT SUBROUTINE IS TO PERFORM ALL KEY FIELD RECORD COMPARISON

TYPE	LABEL	OPERATION	OPERAND	DESCRIPTION
SORT FILE DEFINITION (cont.)	ptn	MR$PRM (cont.)	FIN = symbol	DEFINES ADDRESS TO WHICH CONTROL IS RETURNED AFTER ALL SORTED RECORDS HAVE BEEN DELIVERED TO THE PROBLEM PROGRAM
			IN = symbol	DEFINES THE ADDRESS TO WHICH CONTROL IS RETURNED AFTER THE SORT HAS BEEN INITIALIZED, FOLLOWING EXECUTION OF MR$OPN LINKAGE

			NOCKSM= { device
			(device,device) }
				CAUSES THE CHECKSUM CALCULATION TO BE BYPASSED FOR THE INDICATED DEVICE

			OUT = symbol	DEFINES THE ADDRESS TO WHICH CONTROL IS RETURNED IN THE PROBLEM PROGRAM AFTER RECORDS HAVE BEEN SORTED
			PAD = bytes	PERMITS USER TO AUGMENT THE PARAMETER TABLE BEYOND THE REQUIRED LENGTH
			RCSZ = maximum-bytes	DEFINES SIZE OF FIXED-LENGTH RECORDS OR MAXIMUM SIZE OF VARIABLE-LENGTH RECORDS OF THE DATA TO BE SORTED

			REDO= { CYCLE,cycle-number,from-rec-number,to-rec-number)
			(MERGE,tape label, reel-number to-rec-number) }
				PROVIDES THE CAPABILITY OF REPRODUCING A PART A INTERMEDIATE OUTPUT FILE OR A PART M OUTPUT TAPE

			RESERV = sort-filename	OPTIONAL; USED ONLY FOR A SMALL VOLUME SORT; TAPE UNIT ASSIGNED TO THE SPECIFIED FILE IS USED ONLY DURING PHASES 1 AND 2

			RESUME= { (PASS,recovery-number)
			(CYCLE,cycle-number,from-rec number)
			(MERGE,tape-label,reel-number)
			(FINAL,symbol) }
				USED TO CONTINUE EXECUTING AN INTERRUPTED ACT

			RSOC = symbol	SPECIFIED WHEN THE PROBLEM PROGRAM IS TO COMPARE ALL KEY FIELDS OF ALL RECORDS

TYPE	LABEL	OPERATION	OPERAND	DESCRIPTION
SORT FILE DEFINITION (cont.)		MR$PRM (cont.)	SHARE = sort-filename	TO ALLOW THE PROBLEM PROGRAM TO USE A TAPE UNIT ASSIGNED TO THE SORT SUBROUTINE DURING OPERATIONAL PHASE 1 OF A SORT RUN; USED ONLY FOR A SMALL VOLUME SORT

			STOR = { symbol
			(symbol, number-of-bytes) }
				THE NUMBER OF BYTES TO BE RESERVED IN MAIN STORAGE FOR THE SORT NEED NOT BE SPECIFIED; THE STARTING ADDRESS MUST BE SPECIFIED

			TAPES = { label-type
			(label-type,max-filenumber) }
				SPECIFIES THE NUMBER OF MAGNETIC TAPE UNITS THAT MAY BE USED BY THE SORT AS SCRATCH TAPES

LINKAGE BETWEEN PROBLEM PROGRAM AND SORT SUBROUTINE	[symbol]	MR$ORT		AN INTERFACE MODULE WHICH CONTROLS ALL COMMUNICATION BETWEEN THE PROBLEM PROGRAM AND THE SORT SUBROUTINE

				LIVERED FOR PROCESSING, THE INITIALIZATION LINKAGE CODING MUST BE EXECUTED
	[symbol]	MR$REL		PROVIDES LINKAGE TO RELEASE EACH RECORD TO THE SORT SUBROUTINE, ONE RECORD AT A TIME
	[symbol]	MR$SRT		NOTIFIES THE SORT SUBROUTINE THAT ALL RECORDS TO BE SORTED HAVE BEEN RELEASED
	[symbol]	MR$RET		PROVIDES LINKAGE WHICH RETURNS A SORTED RECORD FROM THE SORT SUBROUTINE, ONE RECORD AT A TIME
	[symbol]	MR$BRK		USED TO TERMINATE OR BREAK CYCLES AT PARTICULAR POINTS DURING PART A PROCESSING OF LARGE VOLUME SORTS
	[symbol]	MR$REC		SUPPLIES THE ADDRESS OF THE FIRST BYTE AND LENGTH OF THE SORT RERUN INFORMATION TABLE

OPERATION	OPERAND	DESCRIPTION
ADDx	name-1 [{ (ALT1) (ALT2) (ALT1,group) (ALT2,group) (CARD) (MCL) /new-name-1 [{ (ALT1) (ALT2) (MCL) }] .ALL [{ (ALT1) (ALT2) (ALT1,group) (ALT2,group) (MCL) }] }] [,name-2...]	USED TO ADD A MODULE OR GROUP OF MODULES TO THE SPECIFIED LIBRARY FILE
CMPx		USED TO COMPRESS A LIBRARY FILE OR PROC GROUP
CORx	progname [{ (ALT1) (ALT2) (ALT1,group) (ALT2,group) }]	USED TO CORRECT LINES OF SOURCE CODE WITHIN SOURCE, COPY, AND PROC LIBRARIES

NOTE: The "x" following the statement denotes a library designator.

OPERATION	OPERAND	DESCRIPTION
CPYx { C S P_n }	group { (ALT 1) (ALT 2) (ALT 1,group) (ALT 2,group) }	USED TO COPY AN ENTIRE LIBRARY FILE OR PROC GROUPS IN SEPARATE FILES
DELx	{ name-1 name-1.ALL ALL } [, { name-2 name-2.ALL } ,...]	USED TO DELETE A MODULE, GROUP OF MODULES, PROC GROUP, OR AN ENTIRE LIBRARY FILE
DISx	{ name-1 name-1.ALL DIR ALL } [, { name-2 name-2.ALL DIR } ,...]	USED TO DISPLAY A MODULE, GROUP OF MODULES, THE DIRECTORY, OR AN ENTIRE LIBRARY FILE
DUMx		USED TO CREATE A TAPE LIBRARY FILE OR PROC GROUP
ENDCARD		USED TO TERMINATE SOURCE STATEMENTS
FILx	LIBn, { I O } { numtracks SAME numtracks 'numcyls }	USED TO INITIALIZE THE SPECIFIED DISC FILE; DEFINE NEW FILES
INPx	LIBn	USED TO SPECIFY THE LIBRARY FILE ON WHICH SOME FUNCTION IS TO BE PERFORMED

OPERATION	OPERAND	DESCRIPTION
INS	line-number	SPECIFIES INSERTION OF SOURCE LINES
OUTx	LIBn	SPECIFIES THE OUTPUT LIBRARY FILE ONTO WHICH THE FILE SPECIFIED BY INP IS TO BE COPIED
PCHx	$\left\{\begin{matrix}\text{name-1}\\ \text{name-1.ALL}\\ \text{ALL}\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{name-2}\\ \text{name-2.ALL}\end{matrix}\right\},\ldots\right]$	CONVERTS A LIBRARY MODULE INTO A PUNCHED CARD OUTPUT DECK
PUDx	$\left\{\begin{matrix}\text{name-1}\\ \text{name-1.ALL}\\ \text{ALL}\end{matrix}\right\}\left[,\left\{\begin{matrix}\text{name-2}\\ \text{name-2.ALL}\end{matrix}\right\},\ldots\right]$	USED TO DISPLAY A MODULE AS PRINTED OUTPUT AND CONVERT A MODULE INTO A PUNCHED CARD OUTPUT DECK
REP	first-line,last-line	SPECIFIES REPLACEMENT OF SOURCE LINES
RNMx	oldname/newname[,oldname/newname,...]	USED TO RENAME A GIVEN ITEM
VER	[level-number] , [update-number]	USED TO CREATE A VERSION NUMBER

OPERATION	OPERAND	DESCRIPTION
ADDx	$\text{name-1}\left[\left\{\begin{matrix}\text{(CARD)}\\ \text{(OBJ)}\\ \text{(ALT)}\end{matrix}\right\}\right]\ \text{[,name-2...]}$	USED TO ADD TO LOAD, RESERVE, COPY OR SOURCE LIBRARY
ADDPn	$\text{name-1}\left[\left\{\begin{matrix}\text{(CARD)}\\ \text{(ALT,group)}\\ \text{(ALT)}\end{matrix}\right\}\right]\text{[,name-2 ...]}$	USED TO ADD TO PROC LIBRARY
ADDL	$\text{TR name-1}\left[\left\{\begin{matrix}\text{(OBJ)}\\ \text{(ALT)}\end{matrix}\right\}\right]\text{[,TR, name-2 ...]}$	USED TO CREATE TRANSIENT ROUTINES AND ADD TO LOAD LIBRARY
$\text{COR}\left\{\begin{matrix}\text{C}\\ \text{S}\\ \text{P}_n\end{matrix}\right\}$	name	USED TO PROCESS SOURCE LINE CORRECTIONS WITHIN THE SOURCE, COPY, AND PROC LIBRARIES
CPY	ALL	USED TO COPY AN ENTIRE LIBRARY STRUCTURE
CPYx	TO,module name-1[,name-2...]	USED TO COPY GROUPS OF MODULES OR ONE OR A NUMBER OF SPECIFIC MODULES
DELx	$\left\{\begin{matrix}\text{TO,module}\\ \text{ALL}\\ \text{name-1[,name-2...]}\end{matrix}\right\}$	USED TO DELETE A GROUP OF MODULES, THE ENTIRE REMAINING PORTION OF A SPECIFIC LIBRARY, OR A SPECIFIC MODULE OR A NUMBER OF SPECIFIC MODULES

NOTE: The "x" following the statement denotes a library designator.

OPERATION	OPERAND	DESCRIPTION
DEL	ALL	USED TO DELETE THE REMAINING PORTION OF THE LIBRARY STRUCTURE
DISx	{ALL DIR}	USED TO DISPLAY EACH LIBRARY WITH ITS HEADER BLOCKS OR THE HEADER BLOCKS ALONE
DISx	name-1[,name-2...]	USED TO DISPLAY MODULES
DIS	{DIR ALL}	USED TO DISPLAY HEADER BLOCKS OF AN ENTIRE LIBRARY STRUCTURE, OR EVERY BLOCK OF A LIBRARY STRUCTURE
ENDCARD		DELIMITS THE SOURCE STATEMENTS WHICH ARE USED FOR SOURCE LINE CORRECTIONS
INS	n_1	USED TO INSERT SOURCE LINES IN A MODULE FOLLOWING THE SPECIFIC NUMBERED LINE
LIB	[IPL][,INIT][,NOPRNT][,NOBJ][,NALT]	USED TO REQUEST A PARTICULAR TYPE OF LIBRARY UPDATE
PCHx	ALL	USED TO CONVERT A LIBRARY TO PUNCHED CARDS
[illegible]	[illegible]	USED TO PUNCH SPECIFIC MODULES WITH HEADER AND ASSOCIATED BLOCKS
PUDx	ALL	USED TO PRINT AND PUNCH MODULES WITH HEADERS
PUDx	name-1[,name-2...]	USED TO DISPLAY AND PUNCH A MODULE
REP	n_1,n_2	USED TO REPLACE SOURCE LINES IN A MODULE STARTING AT LINE n1 AND ENDING AT LINE n2
VER	level-number,update-number	USED TO CREATE A VERSION NUMBER

LINKAGE EDITOR CONTROL STATEMENTS

LABEL	OPERATION	OPERAND	DESCRIPTION
	LOADM	name [,addr}	FIRST CONTROL STATEMENT FOR EACH LOAD MODULE; SPECIFIES NAME AND STARTING ADDRESS OF MODULE TO BE CONSTRUCTED
	LINKOP	$p_1,\ldots,p_n$	SPECIFIES OPTION TO BE USED IN INSTRUCTING A LOAD MODULE; IT MUST IMMEDIATELY FOLLOW LOADM STATEMENT
	INCLUDE	[modulename][($s_1,\ldots,s_n$)][,filename]	INCLUDES SPECIFIED OBJECT MODULES OR SELECTED CONTROL SECTIONS OF SPECIFIED OBJECT MODULES IN CURRENT PHASE OF LOAD MODULE BEING CONSTRUCTED
symbol	EQU	expression	PROVIDES LINKAGE EDITOR WITH VALUE OF AN OTHERWISE UNDEFINED LABEL
	OVERLAY	symbol	INDICATES BEGINNING OF AN OVERLAY PHASE AND DEFINES RELATIVE POSITION OF THE PHASE WITHIN LOAD MODULE STRUCTURE
	MOD	power [,remainder]	ADJUSTS LOCATION COUNTER TO NEXT VALUE WHICH IS GREATER THAN OR EQUAL TO ITS PRESENT VALUE, AND WHICH HAS A SPECIFIED REMAINDER WHEN DIVIDED BY A GIVEN POWER OF 2
	ENTER	expression	PROVIDES ADDRESS OF ENTRY POINT OF THE CURRENT PHASE
	RES	value	RESERVES SPACE IN MAIN STORAGE