

IBM

**Field Engineering Education
Supplementary Course Material**

SYSTEM/360

**Model 25 Microprogram Listing
System/360 Emulator - *E60**

PREFACE

This publication contains a listing of the System/360 emulator microprogram, *Engineering change level and features present do not relate to any specific system, limiting the use of this listing to the classroom.

The listing shown is actual and has not been altered in content in any way.

Routines are in alphabetical sequence as shown in the listing index.

*And is meant for use in the System/360 Model 25 DPS Course 13233 and 53232.

(March 1970)

Address any comments concerning the contents of this publication to:
IBM, Field Engineering Education Media Development Center, Dept 927,
Rochester, Minnesota 55901.

Preliminary Edition

THIS PUBLICATION IS IN A PRELIMINARY STATE OF DEVELOPMENT. ANY CORRECTION OR SUGGESTIONS THAT YOU CAN OFFER FOR THE FORMAL PUBLICATION WILL BE APPRECIATED.

MACHINE TYPE MODEL CORE LOAD *E60

MES/FCSI #

PAGE III

SERIAL # -

MASTER CONTROL

MACHINE STATUS

EC # 128211 SUM CHECK # 45D1 ** SEE *E60 BCHK **

DATE PREPARED 11/08/68

FEATURES PRESENT

2542057	MEM48K	2539229	FLTPT	0854021	STGPROT	2539257	PFR
2539256	BURSTCH	0854031	TIMER	0854019	NATV2540	0854015	NATV1403
0854017	NATV2311	0854023	DIRCTRL	0854095	MULCHSET		

FEATURES NOT PRESENT

2542054	MEM16K	2542056	MEM32K	2539255	BYTECH	2542055	MEM24K
0854030	TIMER50	2532200	COMBASIC				

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
						SYSTEM 360/25 EMULATOR	
	*E60	001	T			AAAB	
	*E60	002	*			AAAC	LOCAL STORAGE MAP CPU
	*E60	003	*			AAAD	EXTERNAL MNEMONIC DEFINITIONS
	*E60	004	*			AAAF	BIT SIGNIFICANCE CHARTS FOR CONTROL WORDS
	*E60	005	*			ALDP	ALTER DISPLAY ROUTINE
	*E60	006	*			BCHK	CHECK SUN
	*E60	007	*			BCPL	RESIDENT CHANNEL CORE LOAD ROUTINE
	*E60	008	*			BDIA	BASIC DIAGNOSTIC...RESIDENT CPU DIAGNOSTIC.
	*E60	009	*			BMCK	MACHINE CK TRAP ROUTINE
	*E60	010	*			BPSW	PSW STORE AND LOADING
	*E60	011	*			BSTP	INSTRUCTION STEP TYPE OUT-INSTRUCTION ADDRESS
	*E60	012	*			BSWI	EXCEPTIONAL CONDITION ROUTINE
	*E60	013	*			BSYS	SYSTEM RESET ROUTINE
	*E60	014	*			BWRP	WRAP TRAP ROUTINES
	*E60	015	*			CBIN	BINARY ARITH.-ADD-SUB-COMPARE
	*E60	016	*			CBRC	BRANCH INSTRUCTIONS
	*E60	017	*			CCOM	CPU COMMON ROUTINES
	*E60	018	*			CCMD	DECIMAL MULT DIVD
	*E60	019	*			CDVD	BINARY DIVIDE ROUTINE
	*E60	020	*			CFAD	FLOATING POINT ADD,SUBTRACT,COMPARE
	*E60	022	*			CFCD	FLOATING POINT COMMON ROUTINES
	*E60	023	*			CFCY	FLOATING POINT INSTRUCTION CYCLES
	*E60	025	*			CFDV	FLOATING POINT DEVIDE
	*E60	027	*			CFHA	FLOATING POINT HALVE
	*E60	028	*			CFLS	FLOATING POINT LOADS AND STORE
	*E60	029	*			CFMD	FLOATING POINT MULT DIV PREPARATION
	*E60	030	*			CFMU	FLOATING POINT MULTIPLY
	*E60	031	*			CICY	INSTRUCTION CYCLES
	*E60	033	*			CLOG	LOGICAL OPERATIONS
	*E60	034	*			CLST	LOAD AND STORE OPERATIONS
	*E60	035	*			CMLT	BINARY MULTIPLY ROUTINE
	*E60	036	*			CMPU	PACK UNPACK AND MOVE WITH OFFSET
	*E60	037	*			CNVR	CONVERT TO DECIMAL AND BINARY
	*E60	038	*			CSFT	SHIFT ROUTINES
	*E60	039	*			CSAS	DECIMAL ADD SUB COMPARE ZERO AND ADD
	*E60	040	*			CSTS	STATUS SWITCHING INSTRUCTIONS
	*E60	041	*			CTRT	TRANSLATE TRANS TEST EDIT EDIT AND MARK
	*E60	042	*			DCHN	CHANNEL 0 INTERRUPT ROUTINE----H.BERKEBILE
	*E60	043	*			DCLA	I/O INSTRUCTIONS
	*E60	044	*			DCLB	CCW FETCH AND CHECK
	*E60	045	*			DCLC	CHANNEL INITIAL SELECTION
	*E60	047	*			DCLD	CHANNEL DATA LOOP
	*E60	049	*			DCLE	CHANNEL LOW PRIORITY TRAP-STATUS ROUTINE
	*E60	050	*			DCLH	CHANNEL HALT I/O
	*E60	052	*			DCLL	INITIAL PROGRAM LOAD
	*E60	054	*			DCLR	CHANNEL ERROR ROUTINE
	*E60	055	*			DCLT	CHANNEL TEST I/O AND I/O INTERRUPT
	*E60	057	*			DCOM	COMMON I/O ROUTINES
	*E60	059	*			DMCS	1403 MULTIPLE CHARACTER SET ROUTINE
	*E60	061	*			DPTC	1403 COMMAND DECODE+CONSTANTS AND AUX TABLES
	*E60	063	*			DPTQ	1403 REQUEST AND INTERRUPT BUFFER ROUTINE
	*E60	064	*			DPTR	1403 READ AND WRITE-CHANNEL END AND CARR.CTRL
	*E60	065	*				

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		*E60 066	*		DPTS	1403 START I/O-NO ACT-NO SEC+SENSE COMM	
		*E60 067	*		DPTT	1403 TEST I/O,HALT I/O,START I/O-SEC. ON	
		*E60 068	*		DYPE	1052 TYPE ROUTINES	
		*E60 070	*		ECCL	PUNCH COLUMN BINARY AND DATA CHAIN	
		*E60 071	*		EPCH	PUNCH SIO CMD DECODE & CHAINING	
		*E60 072	*		EPXF	PUNCH TRANSFER-EBCDIC TO ROW FORM -& CHAINING	
		*E60 073	*		ERCX	XFER FROM COLUMN FORM BUFFER-RO-TO CCW ADDRESS	
		*E60 074	*		ERDR	READER START I/O COMMAND DECODE AND CHAINING	
		*E60 076	*		ERRQ	READER REQUEST MODE AT IO	
		*E60 077	*		ETRP	NATIVE PUNCH TRAP ROUTINE(PUNCH CHECK ANALYSIS)	
		*E60 078	*		ETTR	NATIVE READER TRAP ROUTINE(READER CHECK ANALYSIS)	
		*E60 080	*		EXFR	READER XFER OF ROW IMAGE TO COLUMN BINARY FORM	
		*E60 082	*		FILE	2311 FEAT., ENTRY FOR SIO, TIO , HIO	
		*E60 084	*		FILT	2311 FEAT., TRAP ROUTINE	
		*E60 086	*		FILX	2311 FEAT., OP VERIVICATION & EXECUTION	
		*E60 087	*		FINT	2311 FEAT., INTERRUPT ROUTINE	

* AAAB *

AUX STORAGE -- MODULE 0

* AAAB *

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0X	*****															
*	* G.P. REG 0 * * FLOATING POINT REG. 0 * 0X															
*	*-----*															
1X	* G.P. REG 1 * * H * 1X															
*	*-----*															
2X	* G.P. REG 2 * * 2311 * FLOATING POINT REG. 2 * 2X															
*	*-----*															
3X	* G.P. REG 3 * * J * 3X															
*	*-----*															
4X	* G.P. REG 4 * * 1052 * FLOATING POINT REG. 4 * 4X															
*	*-----*															
5X	* G.P. REG 5 * * ALTER/DISPLAY * K * 5X															
*	*-----*															
6X	* G.P. REG 6 * * MESSAGE * FLOATING POINT REG. 6 * 6X															
*	*-----*															
7X	* G.P. REG 7 * * L * 7X															
*	*-----*															
8X	* G.P. REG 8 * * -R- * K0 * K1 * K2 * K3 * 8X															
*	*-----*															
9X	* G.P. REG 9 * * -R- * K4 * K5 * K6 * K7 * 9X															
*	*-----*															
AX	* G.P. REG A * * * K8 * K9 * KA * KB * AX															
*	*-----*															
BX	* G.P. REG B * * A * B * C * D * KC * KD * KE * KF * BX															
*	*-----*															
CX	* G.P. REG C * * P * M * F * 1403 UCW * CX															
*	*-----*															
DX	* G.P. REG D * * * M * * 2540 RDR UCW * DX															
*	*-----*															
EX	* G.P. REG E * * * M * * 2540 PCH UCW * EX															
*	*-----*															
FX	* G.P. REG F * * E * N * M * G * 1052 UCW * FX															

K-ADDRESSABLE AREA

- K0-88- CHANNEL 1 INTERRUPT BUFFER
- K1-8A- STANDARD INTERFACE, NEXT CCW ADDRESS
- K2-8C- CHANNEL UNIT ADDRESS BUFFER
- K3-8E- STATUS/ACTIVE BYTE FOR 2311 OR CHANNEL 1
- K4-98- 2311 NEXT CCW ADDRESS
- K5-9A- 2311 SENSE OR PREVIOUS OP AND MASK
- K6-9C- 2311 SENSE OR FILE ADDRESS
- K7-9E- DIAGNOSTICS, ALTER/DISPLAY BAL BACKUP
- K8-A8- SYSTEM MASK -A9- CPU KEY AND AMWP
- K9-AA- EXECUTE INSTRUCTION COUNTER ---I REGISTER BACKUP
- KA-AC- U REGISTER BACKUP Q FLTPT SAVE DURING INSTRUCTIONS
- KB-AE- G REGISTER BACKUP
- KC-B8- P REGISTER BACKUP
- KD-BA- CHANNEL 0 INTERRUPT BUFFER
- KE-BC- ADDRESS OF STRAIGHT MULT/DVD , FLTPT SAVE 2540 REGS BACKUP
- KF-BE- ADDRESS OF SKEWED MULT/DVD , FLTPT SAVE OR CAW KEY. 2540 REGS BACKUP

- A----1403 UNIT ADDRESS
- B----READER UNIT ADDRESS
- C----PUNCH UNIT ADDRESS
- D----1052 UNIT ADDRESS
- E----H1 SAVE
- F----1403 PCCL
- G----1052 SENSE
- N----2311 ADDRESS
- H----MULT/DVD TABLE (X1) AND ALT/DIS REGS BACKUP
- J----MULT/DVD TABLE (X4) AND ALT/DIS BACKUP + FLTPT SAVE
- K----MULT/DVD TABLE (X16)
- L----MULT/DVD TABLE (X64), FLTPT SAVE
- M----NATIVE KEY KKKK0000
- P----COMMU. Q-EXIT POINTER
- R----BURST CHANNEL BUFFERED DEVICE ADDRESSES.

***** STANDARD DEVICE ADDRESSES *****

* NATIVE	LOC	* BURST CHANNEL	LOC*
* 1403-----0E	B4	* 1443 OR 1445-----0B	84 *
* 2540 RDR---0C	B5	* 2540 RDR-----0C	85 *
* 2540 PCH---0D	B6	* 2540 PCH-----0D	86 *
* 1052-----1F	B7	* 1403-----0E	87 *
* 2311-----9X	F5	* 1404 OR 2ND 1403---0F	94 *
*		* 2520-----15	95 *

* THE TEXT PRECEEDING THE -BCPL- ROUTINE CONTAINS *
* INFORMATION ON PUNCHING CARDS TO RECONFIGURATE THE *
* SYSTEM TO OTHER THAN THE STANDARD ASSIGNMENTS *

AUXILIARY STORAGE ASSIGNMENTS

0	4	8	C	F	MODULE		

0	*			*			
	*	CPU AREA		*	0	MODULE 0 DEFINED IN AAAB ROUTINE	
	*			*			
F	*	-----*					
0	*			*			
	*	PUNCH ROW IMAGE (1X00-1X77)		*	1	MODULE 1 C AND D NOTED IN TABLE ARE DEFINED IN COMMENTS PRECEEDING THE ERDR ROUTINE. C AT 10F0 D AT 10F1 B, E, AND A NOTED IN TABLE ARE DEFINED IN TEXT PRECEEDING THE EPCH ROUTINE. B AT 10F2, E AT 10F4, A AT 10F5	
	*			*			
	*	PFR ROW IMAGE (1X78-1XEF)		*			
F	*	C D B E A * READ ERROR LOGOUT		*			
	*			*			
0	*			*			
	*	MPX UCWS		*	2	MODULE 2 DEFINED IN FE MAINTENANCE MANUAL	
	*			*			
F	*	-----*					
0	*			*			
	*	READER TRANSLATE TABLE		*	3	MODULE 3 DEFINED AT END OF THE ERCX ROUTINE.	
	*			*			
F	*	-----*					
0	*			*			
	*	PUNCH CHECK (4X0C-4X84)		*	4	MODULE 4 WORKING AREA	
	*	PFR READ (4X00-4X9F)		*			
	*	READER ROW IMAGE (4XA0-5X17)		*			
F	*	-----*					
0	*			*			
	*			*			
	*	READER READOUT (5X18-5XB7)		*	5	MODULE 5 1052 TRANSLATE TABLE DEFINED AT END OF THE DYPE ROUTINE.	
	*			*			
	*	1052 TRANSLATE TABLE (5XB8-5XF5)		*			
F	*	* PUNCH ERROR LOGOUT		*			
	*			*			
0	*			*			
	*	PUNCH TRANSLATE TABLE		*	6	MODULE 6 DEFINED AT END OF THE EPXF ROUTINE. PUNCH TRANSLATE FOR 24K IN MODULE 8.	
	*			*			
F	*	-----*					
0	*			*			
	*	PRINTER TRANSLATE TABLE		*	7	MODULE 7 DEFINED AT END OF THE OPTC ROUTINE. PRINTER TRANSLATE TABLE FOR 24K IN MODULE 9.	
	*			*			
F	*	*****					

* AAAC *

360 MODE LOCAL STORAGE ZONES AND ASSIGNMENTS

* AAAC *

```

*****
*   *   *   ZONE 0   **   ZONE 1   **   ZONE 4   **   ZONE 5   **   ZONE 6   **   ZONE 7   *
* AS/BS * NAME *   CPU   **   2311   **   BACK -UPS **   COMMUNICATIONS **   2540   **   CHANNEL   *
* FIELD *   *   *   **   *   **   *   **   *   **   *   **   *   **   *
*****
*   *   *   *   **   *   **   *   **   *   **   *   **   *   **   *
* 0000 * U0 * 1ST OPERAND **   **   LEVEL 1 **   LCW ADDR HIGH **   FORM   **   COUNT   *
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 0001 * U1 * STORAGE ADDR. **   **   BACK-UP **   LCW ADDR LOW **   ADDRESS **   COUNT   *
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 0010 * V0 * 2ND OPERAND **   **   **   **   **   **   **   **   **   **   *
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 0011 * V1 * STORAGE ADDR **   **   **   **   **   **   **   **   **   **   *
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 0100 * G0 * OP CODE **   **   **   **   **   **   **   **   **   **   *
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 0101 * G1 * 2ND BYTE INST. **   **   **   **   **   **   **   **   **   **   *
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 0110 * D0 * DATA **   **   **   **   **   **   **   **   **   **   *
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 0111 * D1 * HIGH 8 OF 24 BIT **   **   **   **   **   **   **   **   **   **   *
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
*   *   *   * 2ND OPERAND ADDR**   **   **   **   **   **   **   **   **   **   *
*****
***** ZONE 4 THRU 7 **
*****-----**
* 1000 * I0 * INST CTR HIGH ** CCW COUNT HIGH ** CPU BAL **
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 1001 * I1 * INST CTR LOW ** CCW COUNT LOW ** BACK-UP **
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 1010 * I0 * ** DATA ADDR HIGH ** SPARE **
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 1011 * I1 * GP REG ADDRESS ** DATA ADDR LOW ** SPARE **
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 1100 * P0 * CONDITION CODE ** GEN. PURPOSE ** LEVEL 1 **
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 1101 * P1 * PROG MASK, AMWP ** GEN. PURPOSE ** WORKING AREA **
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 1110 * H0 * DATA ** CCW ADDR HIGH ** LEVEL 1 **
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
* 1111 * H1 * HIGH 8 OF 24 BIT ** CCW ADDR LOW ** WORKING AREA **
*-----*-----*-----*   **   **   **   **   **   **   **   **   **   **   *
*   *   *   * INST CTR OR 1ST **   **   **   **   **   **   **   **   **   **   *
*   *   *   * OPERAND ADDR. **   **   **   **   **   **   **   **   **   **   *
*****

```

NOTE 1-- FOR BURST CHANNEL, THIS REGISTER CONTAINS THE UNIT ADDRESS.

FOR BYTE CHANNEL, THIS REGISTER CONTAINS THE UCW ADDRESS.

... THIS AREA OF ZONE 4 IS COMMON TO ZONES 5, 6, AND 7.

EXTERNAL MNEMONIC	DEFINITION	DISPLAYABLE FACILITY	SWITCH C	SWITCH D	ACCESSED BY SET/RESET WORD ONLY	* AAAD *
BA	BRANCH CONDITIONS	ALL MODES EXT TO CPU	YES	---	7	
BB	SOFT STOP BRANCH CONDITIONS	CPU MODE EXT TO CPU	YES	CPU	F	
BC	EXTERNAL FACILITY	ALL MODES	---	---	---	YES
CADR	COMM ADAPTER DIAGNOSTIC REG	COMM MODE CPU TO EXT	---	---	---	
CCTRL	START-STOP CONTROL	COMM MODE	---	---	---	YES
CHI	COUNTER 1 HIGH IN- DIAGNOSTIC	2311 MODE EXT TO CPU	YES	2311	8	
CLI	COUNTER 1 LOW IN - DIAGNOSTIC	2311 MODE EXT TO CPU	YES	2311	9	
CPF	READ DIRECT ENABLE	CPU MODE	---	---	---	YES
CSETF	START STOP/SYNC	COMM MODE	---	---	---	YES
DAIN	DATA IN	COMM MODE EXT TO CPU	YES	COMM	A	
DAOUT	DATA OUT	COMM MODE CPU TO EXT	---	---	---	
DASI	DISK ATTACHMENT STATUS IN	2311 MODE EXT TO CPU	YES	2311	5	
DIAB	DIAGNOSTIC REG	2311 MODE	---	---	---	YES
DIAC	DIAGNOSTIC REG	2311 MODE	---	---	---	YES
DILIN	DIAL IN	COMM MODE EXT TO CPU	YES	COMM	E	
DILOUT	DIAL OUT	COMM MODE CPU TO EXT	---	---	---	
DR	DIAGNOSTIC REGISTER	ALL MODES	YES	CPU	C	NOTE 1
DS	DISK STATUS	2311 MODE EXT TO CPU	YES	2311	E	
DYN	DYNAMIC CONDITION REGISTER	ALL MODES	YES	---	4	
FBI	FILE BUS IN	2311 MODE EXT TO CPU	YES	2311	1	
FBO	FILE BUS OUT	2311 MODE CPU TO EXT	---	---	---	
FEBO	FILE 1400 EMULATOR BUS OUT	2311 MODE CPU TO EXT	---	---	---	
FFI	FILE FLAGS IN	2311 MODE EXT TO CPU	YES	2311	D	
FFO	FILE FLAGS OUT	2311 MODE CPU TO EXT	---	---	---	
FGA	FILE GATED ATTENTION	2311 MODE EXT TO CPU	YES	2311	C	
FIA	FILE INFORMATION	2311 MODE	---	---	---	YES
FIB	FILE INFORMATION	2311 MODE	---	---	---	YES
FIC	FILE INFORMATION	2311 MODE	---	---	---	YES
FOB	FILE OUT BUS-DIAGNOSTIC	2311 MODE EXT TO CPU	YES	2311	3	
FOP	FILE OP REGISTER	2311 MODE	YES	2311	F	
GA	CHANNEL SIGNALS	CHAN MODE	---	---	---	YES
GB	CHANNEL SIGNALS	CHAN MODE	---	---	---	YES
GB/IN	CHANNEL BUS IN	CHAN MODE EXT TO CPU	YES	CHNL	F	
GB/OUT	CHANNEL BUS OUT	CHAN MODE CPU TO EXT	---	---	---	
GC	CHANNEL SIGNALS	CHAN MODE	---	---	---	YES
GD	CHANNEL DIAGNOSTIC REGISTER	CHAN MODE EXT TO CPU	YES	CHNL	E	
GS	CHANNEL BRANCH CONDITIONS	CHAN MODE EXT TO CPU	YES	CHNL	C	
GSTAT	GENERAL STATUS	COMM MODE EXT TO CPU	YES	COMM	F	
GT	CHANNEL BRANCH CONDITIONS	CHAN MODE EXT TO CPU	YES	CHNL	D	
JA	DIRECT CONTROL-2 TIMING	CPU MODE CPU TO EXT	---	---	---	
JI	DIRECT CONTROL IN	CPU MODE EXT TO CPU	YES	CPU	8	
JO	DIRECT CONTROL-1	CPU MODE CPU TO EXT	---	---	---	
LACON	LINE ADAPTER CONDITIONS	COMM MODE EXT TO CPU	YES	COMM	C	
LADR	LINE ADAPTER DIAG REGISTER	COMM MODE CPU TO EXT	---	---	---	
LAIN	LINE ADDRESS IN	COMM MODE EXT TO CPU	YES	COMM	B	
LAOUT	LINE ADDRESS OUT	COMM MODE CPU TO EXT	---	---	---	
LASTAT	LINE ADAPTER STATUS	COMM MODE EXT TO CPU	YES	COMM	D	

EXTERNAL MNEMONIC	DEFINITION	DISPLAYABLE FACILITY	SWITCH C	SWITCH D	ACCESSED BY SET/RESET WORD ONLY	* AAAD *
MC	ERROR REGISTER	CPU MODE EXT TO CPU	YES	CPU	E	
MMSK	MICROPROGRAM MASK REGISTER	ALL MODES	YES (0-7)	---	6	NOTE 2
MODE	MODE REG (LS AND EXT ADDR CTRL)	ALL MODES				NOTE 3
MS	MODULE SELECT REGISTER	2311 MODE CPU TO EXT	---	---	---	
MW	MACH CHK, WAIT STATE LATCHES	CPU MODE CPU TO EXT	---	---	---	
P	PUNCH SIGNALS	2540 MODE	---	---	---	YES
PARCK	COMMUNICATIONS PARITY CHECK	COMM MODE CPU TO EXT	---	---	---	
PCCL	PRINT CHAR COUNTER LENGTH	1403 MODE CPU TO EXT	---	---	---	
PO	PUNCH DATA OUT (READER)	2540 MODE CPU TO EXT	---	---	---	
PR	1403 PLBAR DATA OUT	1403 MODE CPU TO EXT	---	---	---	
PRA	PRINTER SIGNALS	1403 MODE	---	---	---	YES
PRB	PRINTER SIGNALS	1403 MODE	---	---	---	YES
PRC	1403 CARRIAGE DATA OUT	1403 MODE CPU TO EXT	---	---	---	
PRD	1403 DIAGNOSTIC CONDITIONS	1403 MODE EXT TO CPU	YES	1403	F	
PRI	1403 PLB DATA IN	1403 MODE EXT TO CPU	YES	1403	B	
PRO	1403 PLB DATA OUT	1403 MODE CPU TO EXT	---	---	---	
PRS	SENSE/STATUS CONDITIONS	1403 MODE EXT TO CPU	YES	1403	E	
PRT	PLBAR DATA IN	1403 MODE EXT TO CPU	YES	1403	A	
PS	PUNCH BRANCH CONDITIONS	2540 MODE EXT TO CPU	YES	2540	F	
R	READER SIGNALS	2540 MODE	---	---	---	YES
RP	2540 SIGNALS	2540 MODE	---	---	---	YES
RP1	READER/PUNCH DATA IN 1	2540 MODE EXT TO CPU	YES	2540	B	
RP2	READER/PUNCH DATA IN 2	2540 MODE EXT TO CPU	YES	2540	A	
RPD	2540 SIGNALS DIAG	2540 MODE	---	---	---	YES
RPD1	DIAGNOSTIC R/P CONDITIONS 1	2540 MODE EXT TO CPU	YES	2540	8	
RPD2	DIAGNOSTIC R/P CONDITIONS 2	2540 MODE EXT TO CPU	YES	2540	9	
RPS	READER/PUNCH BR CONDITIONS	2540 MODE EXT TO CPU	YES	2540	E	
RS	READER BRANCH CONDITIONS	2540 MODE EXT TO CPU	YES	2540	D	
S	STATUS REGISTER	ALL MODES	YES	NOT 2311	5	NOTE 4
SDI	SERIALIZER/DERIALIZER IN DIAG	2311 MODE EXT TO CPU	YES	2311	B	
SM	SYSTEM MASK	CPU MODE CPU TO EXT	---	---	---	
STPO	STORAGE PROTECT KEY	ALL MODES	YES		SEE NOTE 5	
STP1	STORAGE PROTECT STACK	ALL MODES (NOT 2311)	YES		SEE NOTE 5	
SWAB	CONSOLE ADDRESS SWITCHES A-B	CPU MODE EXT TO CPU	---	---	---	
SWCD	CONSOLE ADDRESS SWITCHES C-D	CPU MODE EXT TO CPU	---	---	---	
TA	1052 SIGNALS	1052 MODE	---	---	---	YES
TC	TERMINATING CONDITIONS	2311 MODE EXT TO CPU	YES	2311	A	
TD	PRKB DIAGNOSTIC REGISTER	1052 MODE EXT TO CPU	YES	PRKB	D	
TE	PRKB DATA OUT	1052 MODE CPU TO EXT	---	---	---	
TGRI	TAG REGISTER IN	2311 MODE EXT TO CPU	YES	2311	0	
TGRO	TAG REGISTER OUT	2311 MODE CPU TO EXT	---	---	---	
TI	1052 DATA IN	1052 MODE EXT TO CPU	YES	PRKB	A	
TIM	TIMER COUNT	CPU MODE EXT TO CPU	YES	CPU	A	
TR	1052 TILT/ROTATE REGISTER	1052 MODE EXT TO CPU	YES	PRKB	B	
TT	PRKB BRANCH CONDITIONS	1052 MODE EXT TO CPU	YES	PRKB	E	
TU	PRKB BRANCH CONDITIONS	1052 MODE EXT TO CPU	YES	PRKB	F	
XINT	EXTERNAL INTERRUPT	CPU MODE EXT TO CPU	YES	CPU	9	

* AAAD *

NOTE 1 THE DR REGISTER IS SET BY THE SET/RESET WORD. EXECUTION OF A BRANCH WORD OR A RETURN WORD, WITH DR BIT 7 ON, SETS THE DIAGNOSTIC BRANCH LATCH.

NOTE 2 THE MMSK REGISTER 0-9 IS SET OR RESET BY THE SET/RESET WORD. FOR OTHER WORD ACCESSES AND DISPLAY, ONLY BITS 0-7 CAN BE ADDRESSED.

NOTE 3 THE MODE REGISTER IS SET BY THE SET/RESET WORD. THE ACTUAL BIT STRUCTURE OF THE MODE REGISTER IS NOT ALWAYS WHAT IS DISPLAYED IN THE CONSOLE INDICATORS LABELED MODE/ZONE REG. THE MODE BIT DECODE IS DISPLAYED IN BITS 2, 3, 4, AND THE LS ZONE BIT DECODE IS DISPLAYED IN BITS 5, 6, AND 7. THESE COULD BE THE FORCED DECODES CAUSED BY AN MMSK BIT BEING SET.

** REFER TO DIAGRAMS 4-14 AND 4-32 IN THE -FEMDM- FOR DISPLAY DETAILS.

NOTE 4 THE S-REGISTER IS SET OR RESET BY THE SET/RESET WORD. IT CAN ALSO BE ACCESSED BY A BRANCH WORD. BITS 0-6 CAN BE DISPLAYED DIRECTLY BUT BIT 7 OF THE DISPLAY INDICATES THE STATUS OF THE S7 BRANCH CONDITION LINE. TO DISPLAY THE ACTUAL CONDITION OF THE S7 LATCH, THE BA FACILITY MUST BE DISPLAYED AND BIT 0 WILL INDICATE THE S7 LATCH STATUS.

NOTE 5

TO DISPLAY STP1 FOR MPX OPERATION--

1. SET CONSOLE SWITCHES A, B, C, AND D TO THE AUXILIARY STORAGE ADDRESS OF THE MPX UCW ASSOCIATED WITH THE I/O DEVICE.
2. SET MODE SWITCH TO AS DATA.
3. PRESS AND RELEASE THE DISPLAY KEY. (M-REG SET TO ADDRESS IN SW, S ABCD).
4. SET MODE SWITCH TO ALU/EXT.
5. SET CONSOLE SWITCHES C, D TO 03.
6. PRESS AND HOLD THE DISPLAY KEY. THE ASSOCIATED I/O PROTECT KEY IS DISPLAYED IN THE BYTE 1 INDICATORS.

TO DISPLAY STP1 FOR SELECTOR CHANNEL KEY

1. SET CONSOLE SWITCHES A, B, C, AND D TO AUX ADDRESS XX88. THE REST OF THE DISPLAY PROCEEDURE IS THE SAME AS THE MPX STP1.

TO DISPLAY STP1 FOR CPU STORAGE KEY

1. SET CONSOLE SWITCHES A, B, C, AND D TO A MAIN STORAGE ADDRESS ASSOCIATED WITH A PARTICULAR STORAGE KEY.
2. SET MODE SWITCH TO MS DATA. THE REST OF THE DISPLAY PROCEEDURE IS THE SAME AS THE MPX STP1.

TO DISPLAY THE FILE STORAGE KEY (WHICH IS NOT IN THE STP1 STACK) DO THE FOLLOWING--

1. SET SWITCHES C, D TO 12.
2. SET MODE SWITCH TO ALU/EXT.
3. PRESS AND HOLD THE DISPLAY KEY. THE FILE KEY IS DISPLAYED IN THE BYTE 1 INDICATORS.

TO DISPLAY STP0

1. SET SWITCH C TO MODE DESIRED--
C=0=CPU MODE (Q REG)
C=1=2311 MODE (FQ REG)
C=5=COMM MODE (HQ REG)
C=7=CHAN MODE (GQ REG)
2. SET SWITCH D TO 2.
3. SET MODE SWITCH TO ALU/EXT.
4. PRESS AND HOLD THE DISPLAY KEY.


```

*****
* WORD * FUNCTION *
* TYPE * DECODE * A-SOURCE FIELD * K FIELD * FUNCTION * WORD *
* 0 1 * 2 3 * 4 5 6 7 * 8 9 10 11 * 12 13 14 * 15 *
*****
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* 0 C * 0 0 * 0000 = U0 * 0000 * 000 - Z=A + KL * 1 *
* * * 0 0 * 0001 = U1 * 0001 * 001 - Z=A,A-,KL * *
* * * 0 0 * 0010 = V0 * 0010 * 010 - Z=A,A-,KH * *
* * * 0 0 * 0011 = V1 * 0011 * 011 - Z=A,A-,KK * *
* * * 0 0 * 0100 = G0 * 0100 * 100 - Z=A + KH * *
* * * 0 0 * 0101 = G1 * 0101 * 101 - Z=A,OE,KL * *
* * * 0 0 * 0110 = D0 * 0110 * 110 - Z=A,OE,KH * *
* * * 0 0 * 0111 = D1 * 0111 * 111 - Z=A,OE,KK * *
* *-----* * *-----* *
* * 0 1 * 1000 = I0 * 1000 * 000 * *
* * 0 1 * 1001 = I1 * 1001 * 001 - A=A,A-,KL * *
* * 0 1 * 1010 = T0 * 1010 * 010 - A=A,A-,KH * *
* * 0 1 * 1011 = T1 * 1011 * 011 - A=A,A-,KK * *
* * 0 1 * 1100 = P0 * 1100 * 100 * *
* * 0 1 * 1101 = P1 * 1101 * 101 - A=A,OE,KL * *
* * 0 1 * 1110 = H0 * 1110 * 110 - A=A,OE,KH * *
* * 0 1 * 1111 = H1 * 1111 * 111 - A=A,OE,KK * *
* *-----* * *-----* *
* * 1 0 * * 000 * *
* * 1 0 * * 001 - A=0,OR,KL * *
* * 1 0 * * 010 - A=0,OR,KH * *
* * 1 0 * * 011 - A=0,OR,KK * *
* * 1 0 * * 100 * *
* * 1 0 * * 101 - A=A + KL * *
* * 1 0 * * 110 - A=A + KH * *
* * 1 0 * * 111 - A=A + KK * *
* *-----* * *-----* *
* * 1 1 * * 000 * *
* * 1 1 * * 001 - A=A,OR,KL * *
* * 1 1 * * 010 - A=A,OR,KH * *
* * 1 1 * * 011 - A=A,OR,KK * *
* * 1 1 * * 100 - A=0 - KH * *
* * 1 1 * * 101 - A=A - KL * *
* * 1 1 * * 110 - A=A - KH * *
* * 1 1 * * 111 - A=A - KK * *
*****

```

THE FUNCTION DECODE IS A COMBINATION OF CONTROL WORD BITS 2, 3, 12, 13, AND 14.

THE -A- SYMBOL IN THE FORMULAS UNDER COLUMNS 12, 13, AND 14 CAN BE ANY OF THE LOCAL STORAGE BYTE SOURCES LISTED UNDER THE A-SOURCE FIELD.

BIT SIGNIFICANCE CHART FOR THE ARITH CONSTANT WORD (WORD TYPE 1)

WORD TYPE	STORAGE CONTROL	DATA REGISTER ADDRESS	ADDRESS REGISTER OR K FIELD	MODIFIER CONTROL	WORD TYPE																																												
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																																		
0	1	00=READ CONTROL 01=READ AUX OR PROGRAM 10=STORE CNTL 11=STORE AUX OR PROGRAM	THIS FIELD MAY ADDRESS LOCAL STORAGE FOR A BYTE OR HALFWORD. BYTE SELECTION OF LOCAL STORAGE IS LIMITED TO THE ODD ADDRESSES. HALFWORD SELECTION IS LIMITED TO EVEN ADDRESSES. EXTERNAL FACILITY ADDRESSING IS ALWAYS DONE IN BYTE MODE, THEREFOR ONLY THE EXTERNALS WITH ODD ADDRESSES CAN BE ACCESSED.	000 - NO UPDATE 001 - DIRECT ADDRESSING 010 - NO ACCESS, PLUS UPDATE. 011 - NO ACCESS, MINUS UPDATE. 100 - LOCAL STORAGE DATE REG, ACCESS, PLUS UPDATE. 101 - LOCAL STORAGE DATE REG, ACCESS, MINUS UPDATE. 110 - EXTERNAL DATA REG, ACCESS, PLUS UPDATE. 111 - EXTERNAL DATA REG, ACCESS, MINUS UPDATE.	0																																												
BIT 11 DETERMINES IF THE AUXILIARY STORAGE AREA OR THE PROGRAM STORAGE AREA IS ACCESSED FOR BIT 2,3 DECODES OF 01 AND 11. THIS IS ONLY TRUE WHEN INDIRECTLY ADDRESSING.																																																	
WHEN NO ACCESS TO STORAGE IS DESIGNATED BY THE MODIFIER CONTROL FIELD, BITS 2 AND 3 INDICATE THE UPDATE VALUE.																																																	
BITS 2,3	VALUE	LOCAL STORAGE DECODES																																															
00	0	0000 - U0	0	01	01	01	01	0	11	11	11																																						
01	+ OR - 1	0001 - U1	1	01	01	01	01	0	11	11	11																																						
10	+ OR - 2	0010 - V0	0	11	11	11	11	0	01	01	01																																						
11	+ OR - 2	0011 - V1	1	01	01	01	01	0	11	11	11																																						
		0100 - G0	FOR DIRECT ADDRESSING																																														
		0101 - G1	THIS FIELD CONTAINS BIT																																														
		0110 - D0	CODES THAT FORCE THE ADDRESS																																														
		0111 - D1	REGISTER M1 TO SPECIFIC VALUES*																																														
		1000 - I0	THESE VALUES ARE-																																														
		1001 - I1																																															
		1010 - I0																																															
		1011 - T1																																															
		1100 - P0																																															
		1101 - P1																																															
		1110 - H0																																															
		1111 - H1																																															
ALL EVEN ADDRESSES CAN BE USED AS ADDRESS REGISTERS IN THE FIELD DESIGNATED BY CONTROL WORD* BITS 8,9,10,AND 11.																																																	
<table border="1"> <thead> <tr> <th>BITS 8,9,10,11</th> <th>FORCED M1 VALUES</th> </tr> </thead> <tbody> <tr><td>0000</td><td>88</td></tr> <tr><td>0001</td><td>8A</td></tr> <tr><td>0010</td><td>8C</td></tr> <tr><td>0011</td><td>8E</td></tr> <tr><td>0100</td><td>98</td></tr> <tr><td>0101</td><td>9A</td></tr> <tr><td>0110</td><td>9C</td></tr> <tr><td>0111</td><td>9E</td></tr> <tr><td>1000</td><td>A8</td></tr> <tr><td>1001</td><td>AA</td></tr> <tr><td>1010</td><td>AC</td></tr> <tr><td>1011</td><td>AE</td></tr> <tr><td>1100</td><td>B8</td></tr> <tr><td>1101</td><td>BA</td></tr> <tr><td>1110</td><td>BC</td></tr> <tr><td>1111</td><td>BE</td></tr> </tbody> </table>																BITS 8,9,10,11	FORCED M1 VALUES	0000	88	0001	8A	0010	8C	0011	8E	0100	98	0101	9A	0110	9C	0111	9E	1000	A8	1001	AA	1010	AC	1011	AE	1100	B8	1101	BA	1110	BC	1111	BE
BITS 8,9,10,11	FORCED M1 VALUES																																																
0000	88																																																
0001	8A																																																
0010	8C																																																
0011	8E																																																
0100	98																																																
0101	9A																																																
0110	9C																																																
0111	9E																																																
1000	A8																																																
1001	AA																																																
1010	AC																																																
1011	AE																																																
1100	B8																																																
1101	BA																																																
1110	BC																																																
1111	BE																																																
NOTE-- WHEN DIRECT ADDRESSING CONTROL STORAGE, MO IS SET TO X3. WHEN DIRECT ADDRESSING AUXILIARY STORAGE, MO IS SET TO 00.																																																	

SIGNIFICANCE CHART FOR THE STORAGE WORD (WORD TYPE 2)

```

*****
* WORD * FUNCTION *
* TYPE * DECODE *
*****
* 0 1 * 2 3 * 4 5 6 7 * 8 9 10 11 * 12 13 14 * 15 *
*****
* * * * *
* 0 1 * 0 0 * 0000 = U0 * 0000 = U0 * 000 * 1 *
* * 0 0 * 0001 = U1 * 0001 = U1 * 001 - B=AXH + BL * *
* * 0 0 * 0010 = V0 * 0010 = V0 * 010 - B=AXL,OR,BH * *
* * 0 0 * 0011 = V1 * 0011 = V1 * 011 - * *
* * 0 0 * 0100 = G0 * 0100 = G0 * 100 - B=A + B * *
* * 0 0 * 0101 = G1 * 0101 = G1 * 101 - B=AH + BL * *
* * 0 0 * 0110 = D0 * 0110 = D0 * 110 - B=AL + BH * *
* * 0 0 * 0111 = D1 * 0111 = D1 * 111 - EXT=B * *
* * * * *
* * 0 1 * 1000 = I0 * 1000 = I0 * * *
* * 0 1 * 1001 = I1 * 1001 = I1 * 000 - B=AX * *
* * 0 1 * 1010 = T0 * 1010 = T0 * 001 - B=AXH * *
* * 0 1 * 1011 = T1 * 1011 = T1 * 010 - B=AXL * *
* * 0 1 * 1100 = P0 * 1100 = P0 * 011 - B=0 (STOP) * *
* * 0 1 * 1101 = P1 * 1101 = P1 * 100 - B=A * *
* * 0 1 * 1110 = H0 * 1110 = H0 * 101 - B=AH * *
* * 0 1 * 1111 = H1 * 1111 = H1 * 110 - B=AL * *
* * 0 1 * * * 111 - B=EXT * *
* * * * *
* * 1 0 * THE FUNCTION DECODE IS A COMBINATION OF CONTROL WORD * 000 - A=A,OE,B * *
* * 1 0 * BITS 2,3,12,13, AND 14. * 001 - A=A + B * *
* * 1 0 * * 010 - A=A,OR,B * *
* * 1 0 * IF THE FUNCTION DECODE IS EQUAL TO 00111, THE A-SOURCE FIELD * 011 - A=A,A,B * *
* * 1 0 * ADDRESSES AN EXTERNAL FACILITY. * 100 - AC=A + B+1 * *
* * 1 0 * IF THE FUNCTION DECODE IS EQUAL TO 01111, THE A-SOURCE FIELD * 101 - AC=A + B * *
* * 1 0 * ADDRESSES AN EXTERNAL FACILITY. * 110 - AC=A + B+C * *
* * 1 0 * * 111 - AC=AL + B+C * *
* * * * *
* * 1 1 * THE SYMBOL -A- USED IN THE FUNCTION DECODE FORMULAS, REFERS * 000 - A=A - B+1 * *
* * 1 1 * TO ANY OF THE LOCAL STORAGE SYMBOLS UNDER THE A-SOURCE FIELD. * 001 - A=A - B * *
* * 1 1 * THE SYMBOL -B- USED IN THE FUNCTION DECODE FORMULAS, REFERS * 010 * *
* * 1 1 * TO ANY OF THE LOCAL STORAGE SYMBOLS UNDER THE B-SOURCE FIELD. * 011 - * *
* * 1 1 * * 100 - AC=A - B+C * *
* * 1 1 * THE SYMBOL -C- REFERS TO THE ADDER CARRY (S3 LATCH) * 101 - AC=0 - B+C * *
* * 1 1 * * 110 - A=A +- B+C * *
* * 1 1 * * 111 - AC=A,D+-,B+C * *
*****

```

BIT SIGNIFICANCE CHART FOR THE MOVE/ARITHMETIC WORD (WORD TYPE 3)

```

*****
* WORD *
* TYPE * REPLACEMENT BITS FOR THE M0-REGISTER * REPLACEMENT BITS FOR THE M1-REGISTER *
* 0 1 * 2 3 4 5 6 7 * 8 9 10 11 12 13 14 * 15 *
*****
* 1 0 *
* * BITS 2 THRU 7 OF THIS CONTROL WORD CONTAIN * BITS 8 THRU 14 CONTAIN THE VALUES THAT ARE GATED TO * 0 *
* * THE VALUES THAT ARE GATED TO BITS 2 THRU 7 OF * BITS 0 THRU 6 OF THE M1-REGISTER WHEN THIS WORD IS *
* * THE M0-REGISTER WHEN THIS WORD IS EXECUTED. * EXECUTED. *
* * * *
* * WHEN THIS WORD IS GATED INTO THE CONTROL * *
* * REGISTER, BIT 5 OF THE CONTROL REGISTER IS * *
* * FORCED TO A 1. THE REPLACEMENT VALUE FOR THE * *
* * M0-REGISTER BIT 5 POSITION IS GATED FROM THE * *
* * STORAGE DATA BUS BIT 5 POSITION. * *
* * * *
* * * *
* * WHEN THIS WORD IS EXECUTED, THE ADDRESS OF THE NEXT SEQUENTIAL * *
* * CONTROL WORD IS STORED IN THE I-REGISTER OF LOCAL STORAGE ZONE 4. * *
*****

```

BIT SIGNIFICANCE CHAR FOR THE BRANCH UNCONDITIONAL WORD (WORD TYPE 4)

```

*****
*   WORD   *   BRANCH   *                      *   REPLACEMENT BITS FOR *   REPLACEMENT BITS FOR *   WORD *
*   TYPE   *   CONTROL  *          A SOURCE FIELD          *   M1-REGISTER          *   M0-REGISTER.        *   TYPE *
*****
*   0   1   *   2       3   *   4       5       6       7   *   8   *   9       10      11 *   12      13      14   *   15   *
*****
*   *         *         *         *         *         *         *         *         *         *         *         *         *         *
*   1   0   *   00= 4 WAY BR. *         BIT 5=0 MEANS STRAIGHT * 0=LS *   BIT 9 CONTAINS THE *   BITS 12, 13, AND 14 *   *
*         *   01= 8 WAY BR   *         A-REG GATING.         * 1=EXT *   REPLACEMENT VALUE FOR *   CONTAIN THE REPLACEMENT * 1   *
*         *   10= 2 WAY BR   *         BIT 5=1 MEANS CROSS  *     *   * BIT 1 OF THE *   VALUES FOR BITS 5, 6, *   *
*         *   {A SOURCE *         A-REG OUTPUT *     *   * M1-REGISTER. *   AND 7 OF THE M0 REG. *   *
*         *   NOT ZERO) *         ----- *     *   * *         *   *         *   *
*         *   11= 16 WAY BR *   WHEN THIS CONTROL WORD IS READ *   * BIT 10 CONTAINS THE *   *         *   *
*         * *         *   INTO THE CONTROL REGISTER, BIT 5 *   * REPLACEMENT VALUE FOR *   *         *   *
*         * *         *   IS FORCED TO 1. THIS RESTRICTS *   * BIT 2 OF THE M1 *   *         *   *
*         * *         *   THE FACILITIES THAT CAN BE *   * REGISTER. *   *         *   *
*         * *         *   ADDRESSED BY THE A-SOURCE FIELD *   * *         *   *
*         * *         *   TO THOSE THAT HAVE AT LEAST BIT *   * BIT 11 CONTAINS THE *   *         *   *
*         * *         *   5=1 IN THEIR A-SOURCE DECODES. *   * REPLACEMENT VALUE FOR *   *         *   *
*         * *         *   BECAUSE BIT 5 IS FORCED IN THE *   * BIT 0 OF THE M1 *   *         *   *
*         * *         *   CONTROL REGISTER, THE TRUE STATUS *   * REGISTER *   *         *   *
*         * *         *   OF BIT 5 IS TAKEN FROM THE *   * *         *   *
*         * *         *   STORAGE DATA BUS OUT, FOR A-REG *   * *         *   *
*         * *         *   GATING. *   * *         *   *
*         * *         *         *         *         *         *         *         *         *         *
*         * *         *   EITHER DECODE *         LOCAL *         *         *         *         *
*         * *         *   IN CONTROL *         STORAGE *         *         *         *         *
*         * *         *   STORAGE. *         REGISTER *         *         *         *         *
*         * *         *         *         *         *         *         *         *         *         *
*         * *         *   0000 OR 0100 *         G0 *         *         *         *         *
*         * *         *   0001 OR 0101 *         G1 *         *         *         *         *
*         * *         *   0010 OR 0110 *         D0 *         *         *         *         *
*         * *         *   0011 OR 0111 *         D1 *         *         *         *         *
*         * *         *   1000 OR 1100 *         P0 *         *         *         *         *
*         * *         *   1001 OR 1101 *         P1 *         *         *         *         *
*         * *         *   1010 OR 1110 *         H0 *         *         *         *         *
*         * *         *   1011 OR 1111 *         H1 *         *         *         *         *
*         * *         *         *         *         *         *         *         *         *         *
*         * *         *   EXTERNAL FACILITIES HAVE THE *         *         *         *         *         *
*         * *         *   SAME ADDRESSING RESTRICTIONS. *         *         *         *         *         *
*         * *         *         *         *         *         *         *         *         *         *
*****

```

BIT SIGNIFICANCE CHART FOR BRANCH ON MASK WORD (WORD TYPE 5)

* WORD *	* BIT *	* A-SOURCE FIELD *	* REPLACEMENT BITS FOR THE M1-REGISTER *	* WORD *											
* TYPE *	* CONTROL *			* TYPE *											
* 0	* 1	* 2	* 3	* 4	* 5	* 6	* 7	* 8	* 9	* 10	* 11	* 12	* 13	* 14	* 15
* 1	* 1	* 00= BIT 0 OR 4	* BIT 5=0 MEANS STRAIGHT	* 0=LS	* 1=EXT	* BITS 9--14 CONTAIN THE REPLACEMENT VALUES					* 0	* TEST FOR 0			
		* 01= BIT 1 OR 5	* A-REG GATING.			* FOR BITS 1--6 OF THE M1-REGISTER.					* 1	* TEST FOR 1			
		* 10= BIT 2 OR 6	* BIT 5=1 MEANS CROSS								* 0				
		* 11= BIT 3 OR 7	* A-REG OUTPUT.								* 1				
		* ----- *													
		* WHEN THIS CONTROL WORD IS READ *													
		* INTO THE CONTROL REGISTER, BIT 5 *													
		* IS FORCED TO 1. THIS RESTRICTS *													
		* THE FACILITIES THAT CAN BE *													
		* ADDRESSED BY THE A-SOURCE FIELD *													
		* TO THOSE THAT HAVE AT LEAST BIT *													
		* 5=1 IN THEIR A-SOURCE DECODES. *													
		* BECAUSE BIT 5 IS FORCED IN THE *													
		* CONTROL REGISTER, THE TRUE STATUS*													
		* OF BIT 5 IS TAKEN FROM THE *													
		* STORAGE DATA BUS OUT, FOR A-REG *													
		* GATING. *													
		* ----- *													
		* EITHER DECODE	* LOCAL												
		* IN CONTROL	* STORAGE												
		* STORAGE	* REGISTER												
		* 0000 OR 0100	* G0												
		* 0001 OR 0101	* G1												
		* 0010 OR 0110	* D0												
		* 0011 OR 0111	* D1												
		* 1000 OR 1100	* P0												
		* 1001 OR 1101	* P1												
		* 1010 OR 1110	* H0												
		* 1011 OR 1111	* H1												
		* ----- *													
		* EXTERNAL FACILITIES HAVE THE *													
		* SAME ADDRESSING RESTRICTIONS. *													
		* ----- *													

BIT SIGNIFICANCE CHART FOR BRANCH DN CONDITION WORD (WORD TYPE 6 OR 7)

ALDP -- CONSOLE PRINTER-KEYBOARD ALTER/DISPLAY

ADDITIONAL INFORMATION REFERENCE --
FEMDM DIAGRAMS 5-77 THROUGH 5-79

DESCRIPTION

THIS ROUTINE ALLOWS MANUAL 1052 ALTER OR DISPLAY OF AUXILIARY STORAGE, CONTROL STORAGE, AND PROGRAM STORAGE. ANY NUMBER OF BYTES CAN BE ALTERED. DISPLAYING, HOWEVER, IS ALWAYS IN MULTIPLES OF EIGHT BYTES.

WHEN THE 'PROCEED' LIGHT COMES ON AFTER PRESSING THE PR-KB ALTER DISPLAY BUTTON, TYPE THE FIRST TWO CHARACTERS AS FOLLOWS --

1ST CHARACTER -- A OR D

A = ALTER

D = DISPLAY

2ND CHARACTER -- A, C, OR P

A = AUXILIARY STORAGE

C = CONTROL STORAGE -- CE MODE ONLY

P = PROGRAM STORAGE

AFTER THE FIRST TWO CHARACTERS, TYPE A FOUR-HEX-DIGIT ADDRESS. THE ROUTINE THEN FORCES A LINE FEED. IF THE OPERATION IS ALTER, THE ROUTINE WAITS FOR THE OPERATOR TO TYPE IN THE DATA TO BE ALTERED. IF THE OPERATION IS DISPLAY, THE ROUTINE STARTS RIGHT AFTER THE LINE FEED, TYPING OUT THE FIRST EIGHT BYTES OF DATA THEN COMING TO A HALT. IF THE DISPLAY OF MORE DATA IS DESIRED, THE OPERATOR CAN PRESS THE 'SPACE' BAR, AND THE NEXT EIGHT SEQUENTIAL BYTES WILL BE DISPLAYED. EITHER OPERATION, ALTER OR DISPLAY, SHOULD BE TERMINATED BY AN EOB.

DESCRIPTION CONTINUED

AT ANY POINT DURING THE PROCESS, THE OPERATOR CAN PRESS THE 'CANCEL' KEY TO TERMINATE THE OPERATION EXCEPT WHILE DATA IS BEING TYPED FOR A DISPLAY OPERATION IN WHICH CASE THE KEYBOARD IS LOCKED. WHEN A 'CANCEL' IS ENCOUNTERED, THE OPERATION RETURNS TO THE BEGINNING OF THE ROUTINE GIVING THE OPERATOR A CHANCE TO TRY AGAIN.

EXCEPT THE FIRST TWO CHARACTERS, ALL THE OTHERS SHOULD BE HEX DIGITS ONLY. BOTH UPPER OR LOWER CASE ARE ALLOWED FOR ALPHABETICAL CHARACTERS. THE FOLLOWING CONDITIONS ARE RECOGNIZED AS ERRORS.

1. A NON-HEX DIGIT TYPED ON THE KEYBOARD.
2. FIRST CHARACTER OTHER THAN A OR C.
3. SECOND CHARACTER OTHER THAN A, C, OR P. THE C IS VALID ONLY IN CE MODE.
4. INVALID STORAGE ADDRESS.

THE ROUTINE SENDS A MESSAGE OF 'INVALID CHAR' OR 'INVALID ADDR', THEN RETURNS TO THE BEGINNING TO GIVE THE OPERATOR ANOTHER TRY.

THIS ROUTINE PROVIDES OFFSETTING WHEN THE ADDRESS IS ODD.

WHEN ONE PASS OF THE OPERATION IS COMPLETED, THE ROUTINE ALWAYS GOES BACK TO THE BEGINNING. THUS, THE OPERATOR CAN DO ADDITIONAL ALTER OR DISPLAY OPERATIONS WITHOUT PRESSING THE ALTER-DISPLAY BUTTON AGAIN, OR CAN PRESS THE 'START' BUTTON ON THE CONSOLE IF HE WISHES TO GO BACK TO THE INTERRUPTED CPU PROCESS.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ALDP 001	T			1052 DOCUMENTARY ALTER DISPLAY	R. C. HUANG 10/24/67
		ALDP 002	*			THE WORD LABELED AS 'DOIT' IN BCPL ROUTINE WITH FIXED CONTROL STORAGE	
		ALDP 003	*			ADDRESS OF 03BC VARIES ACCORDING TO THE TYPE OF ALDP OPERATION -	
		ALDP 004	*		AA	- ALTER AUX.	- STB H1 AS,U+1 - 7F08
		ALDP 005	*		AC	- ALTER CONTROL	- STB H1 CS,U+1 - 6F08
		ALDP 006	*		AP	- ALTER PROGRAM	- STB H1 U+1 - 7F18
		ALDP 007	*		DA	- DISPLAY AUX.	- RDH H AS,U+2 - 5E08
		ALDP 008	*		DC	- DISPLAY CONTROL	- RDH H CS,U+2 - 4E08
		ALDP 009	*		DP	- DISPLAY PROGRAM	- RDH H U+2 - 5E18
		ALDP 010	*				
		ALDP 011	*				
		ALDP 012	*			THE REGISTER USAGES IN THIS ROUTINE ARE -	
		ALDP 013	*			U0,U1 - ADDRESS	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ALDP 014	*			V0	- COUNTER OF NUMBER OF HALF WORDS
		ALDP 015	*			V1	- OUTPUT DATA
		ALDP 016	*			G0	- INPUT DATA
		ALDP 017	*			G1	- COLUMN COUNTER
		ALDP 018	*			D0,D1	- CONTROL WORD AND FLAGS
		ALDP 019	*			H0,H1	- DATA
		ALDP 020	*				
		ALDP 021	ASEQ	ALD7=1C			
161C	26A3	ALDP 022	START			DO=0\$K0A	SET ALLOW START KEY FLAG
161E	2783	ALDP 023				D1=0\$K08	DEFINE ADDR TO BE IN U REG'S
1620	873C	ALDP 024		182	LINEFD	BAL	LINE FEED
1622	8744	ALDP 025		187	READCH1	BAL	READ 1ST CHARACTER
1624	3643	ALDP 026				DO=D0\$K04	DISABLE START KEY
1626	043D	ALDP 027				Z=G0\$K30	MASK 3X IN PTT CODE
1628	E0C3	ALDP 028		041	IS3X	BR IF HZ=0	BR IF 1ST CH. IS 3X
162A	873C	ALDP 029	BADCHR	182	LINEFD	BAL	INVALID CHARACTER MESSAGE
162C	2483	ALDP 030	BADADR			GO=0\$K0B	NEGATIVE GO=F4
162E	2577	ALDP 031				G1=0\$K77	SET G=0B77 TO ADDR MESSAGE
1630	534A	ALDP 032	RDMORE			RDB V1 AS,G-1	READ OUT MESSAGE FROM AUX. 0
1632	874E	ALDP 033		193	TYPCHR	BAL	TYPE MESSAGE OUT
1634	D131	ALDP 034		032	RDMORE	BR IF G1 BIT5=1	BR IF NOT TO CHANGE ROW
1636	D01D	ALDP 035		022	START	BR IF G0 BIT5=1	BR IF MESSAGE COMPLETED
1638	7543	ALDP 036	CHGROW			G1=G1-G0	CHANGE ROW BY ADDING F4 OR F0
163A	F530	ALDP 037		032	RDMORE	BR IF G1 BIT3=0	BR IF TO PRINT 2ND OR 4TH ROW
163C	24F3	ALDP 038				GO=0\$K0F	NEGATIVE GO=F0, & SET DONE FLAG
163E	F330	ALDP 039		032	RDMORE	BR IF D1 BIT7=0	BR IF TO PRINT 'CHAR'
1640	9638	ALDP 040		036	CHGROW	BR	GO TO SUBTRACT ONE ROW
1642	044B	ALDP 041	IS3X			Z=G0\$K04	MASK FOR '0' - DISPLAY
1644	FOCF	ALDP 042		047	AORD	BR IF LZ=0	BR IF 1ST CH. IS 'D'
1646	041B	ALDP 043				Z=G0\$K01	MASK FOR 'A' - ALTER
1648	FOAA	ALDP 044		029	BADCHR	BR IF LZNZ	BR IF 1ST CH. IS NOT 'A'
164A	3625	ALDP 045				D0=D0\$K20	MAKE UP CONTROL WORD FOR ALTER
164C	3613	ALDP 046				D0=D0\$K01	D0=2F
164E	8746	ALDP 047	AORD	188	READCH	BAL	READ 2ND CHARACTER
1650	3645	ALDP 048				D0=D0\$K40	D0=4E FOR DSPLY, D0=6F FOR ALTER
		ALDP 049	*				
		ALDP 050	*				
		ALDP 051	*				
		ALDP 052	*				
		ALDP 053	*				D REG'S ARE USED HERE TO SET UP THE VARIABLE CONTROL WORD. THEY ARE ALSO USED IN THIS ROUTINE AS FLAGS. THE USAGE OF EACH FLAG BIT IS AS FOLLOWS -
		ALDP 054	*				DO BIT 0 - EXPECTING EOB IN 'READCH' BAL ROUTINE
		ALDP 055	*				1 - CHAR. READ REQUIRES PACKING IN 'READCH' BAL ROUTINE
		ALDP 056	*				2 - OPERATION IS ALTER
		ALDP 057	*				3 - OPERATION IS IN AUX OR PROGRAM STORAGE - NOT CONTROL
		ALDP 058	*				4 - NOT FROM 'READCH' BAL ROUTINE
		ALDP 059	*				5 - DISABLE 'START' KEY IN THE SOFT STOP LOOP - BSWI
		ALDP 060	*				6 - VALID CHARACTER IN 'READCH' BAL ROUTINE
		ALDP 061	*				7 - NOT USED, 1 FOR ALTER, 0 FOR DISPLAY
		ALDP 062	*				D1 BIT 0 - ODD ADDRESS
		ALDP 063	*				1 - LOOPING CONTROL 1
		ALDP 064	*				2 - LOOPING CONTROL 2
		ALDP 065	*				3 - OPERATION IS IN PROGRAM STORAGE
		ALDP 066	*				4 - THE OPERATION IS NOT FROM 'BSTP' ROUTINE FOR INSTRUCTION ADDRESS TYPE OUT NOR LOG-OUT
		ALDP 066	*				

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ALDP 067	*			5 - 16TH CHARACTER	
		ALDP 068	*			6 - LOG OUT FOR MACHINE CK OR CE TRAP	
		ALDP 069	*			7 - TO PRINT THE MESSAGE OF 'INVALID ADDR'	
		ALDP 070	*				
1652	042D	ALDP 071				Z=GOBK20	MASK 2ND CHAR FOR 2X IN PTT CODE
1654	E0DC	ALDP 072		076	HINOT2	BR IF HZNZ	BR IF NOT 2X
1656	3715	ALDP 073				D1=D1\$K10	SET CONTROL WORD FOR PROG. STORE
1658	047B	ALDP 074				Z=GOBK07	MASK FOR 'P' - 27
165A	5668	ALDP 075		082	BRBAD	BR	GO TO CHECK ON '7'
165C	043F	ALDP 076	HINOT2			Z=GOBK33	MASK 2ND CH. FOR C(33) OR A(31)
165E	E0AA	ALDP 077		029	BADCHR	BR IF HZNZ	BR IF NOT 3X - INVALID 2ND CHAR
1660	F0E6	ALDP 078		081	POSAUX	BR IF LZNZ	BR IF POSSIBLE AUX STORAGE
1662	E66C	ALDP 079		084	ALLSET	BR IF DO BIT2=0	2ND CHAR IS 'C', BR IF DSPY CTRL
1664	C9ED	ALDP 080		084	ALLSET	BR IF TD BIT4=1	ALTER CTRL, BR IF IN CE MODE
1666	041B	ALDP 081	POSAUX			Z=GOBK01	MASK FOR A(31) - AUX STORAGE
1668	F0AA	ALDP 082	BRBAD	029	BADCHR	BR IF LZNZ	BR IF 2ND CH. NOT P(27) OR A(31)
166A	3615	ALDP 083				DO=DO\$K10	SET CTRL WORD FOR AUX OR PROG
166C	6662	ALDP 084	ALLSET			STH D DC,9C	** STORE THE CTRL WD IN 'DOIT'(039C)
166E	874C	ALDP 085		192	SPACE	BAL	FORCE A SPACE
1670	8746	ALDP 086		188	READCH	BAL	READ THE 1ST ADDR HEX DIGIT
1672	5403	ALDP 087				UO=GOXH	PUT IT IN UO HIGH
1674	8746	ALDP 088		188	READCH	BAL	READ THE 2ND ADDR HEX DIGIT
1676	440D	ALDP 089				UO=GOL+UOH	PUT IT IN UO LOW
1678	8746	ALDP 090		188	READCH	BAL	READ THE 3RD ADDR HEX DIGIT
167A	5413	ALDP 091				U1=GOXH	PUT IT IN U1 HIGH
167C	8746	ALDP 092		188	READCH	BAL	READ THE 4TH ADDR HEX DIGIT
167E	441D	ALDP 093				U1=GOL+U1H	PUT IT IN U1 LOW
1680	873C	ALDP 094		182	LINEFD	BAL	FORCE A LINE FEED
1682	F0CA	ALDP 095		099	HWDCNT	BR IF GO BIT7=0	BR IF ADDR IS EVEN
1684	874C	ALDP 096		192	SPACE	BAL	ADDR IS ODD, OFFSET BY
1686	874C	ALDP 097		192	SPACE	BAL	ONE BYTE
1688	17AD	ALDP 098				D1=D1\$KA0	FLIP LOOP CTRL & ODD ADDR FLAGS
168A	2243	ALDP 099	HWDCNT			VO=0\$K04	SET COUNTER TO 4 HW'S
168C	2515	ALDP 100	COLCNT			G1=0\$K10	SET COLUMN COUNT TO 16
168E	F737	ALDP 104	ONEMOR	148	PROGRM	BR IF D1 BIT3=1	BR IF PROG STORAGE
1690	F61B	ALDP 108		117	EXPEOB	BR IF DO BIT3=1	48K, BR IF AUX, ALL MODULE VALID
1692	00C9	ALDP 112				Z=UO+KCO	MASK FOR ADDR HIGHER THAN 3FFF
1694	F49A	ALDP 114	ADDRCK	117	EXPEOB	BR IF AC=0	BR IF ADDR VALID
1696	3713	ALDP 115	ADRCHK			D1=D1\$K01	TO TYPE 'INVALID ADDR'
1698	562C	ALDP 116		030	BADADR	BR	ADDR OUT OF BOUNDARY
169A	3685	ALDP 117	EXPEOB			DO=DO\$K80	SET 'EXPECTING EOB' FLAG
169C	E63A	ALDP 118		152	DISPLY	BR IF DO BIT2=0	BR IF DISPLAY
169E	8746	ALDP 119		188	READCH	BAL	ALTER, READ 1ST HEX
16A0	54F3	ALDP 120				H1=GOXH	PUT IT IN H1 HIGH
16A2	8746	ALDP 121		188	READCH	BAL	READ 2ND HEX
16A4	44FD	ALDP 122				H1=GOL+H1H	PUT IT IN H1 LOW
16A6	172D	ALDP 123				D1=D1\$K20	INVERT LOOPING CONTROL FLAG
16A8	839C	ALDP 124		BCHK 104	DOIT	BAL	STORE THE BYTE
16AA	E70F	ALDP 125		104	ONEMOR	BR IF D1 BIT2=1	BR IF NO SPACING REQUIRED
16AC	874C	ALDP 126	GOSPC	192	SPACE	BAL	FORCE A SPACE
16AE	25FF	ALDP 127	DECONT			G1=G1*KFF	DECREMENT COLUMN COUNT BY ONE
16B0	C48E	ALDP 128		104	ONEMOR	BR IF ZNZ	BR IF NOT 16 COLUMNS YET
16B2	873C	ALDP 129		182	LINEFD	BAL	END OF ONE LINE, FORCE LINE FEED

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1684	568C	ALDP 130		100	COLCNT	BR	GO TO RESTORE COL. COUNTER
1686	0049	ALDP 148	PROGRM			Z=U0+K40	**48K, MASK FOR PROG ADDR HIGHER THAN B
1688	9694	ALDP 150		114	ADDRCK	BR	
168A	839C	ALDP 152	DISPLY	BCHK 104	DOIT	BAL	READ HALFWORD
168C	1645	ALDP 153				DO=DO*-K40	ALLOW ANY CHAR FOR *SPACE*
168E	C744	ALDP 154		157	ENTRY	BR IF D1 BIT0=0	BR IF ADDR IS EVEN
16C0	1785	ALDP 155				D1=D1*-K80	ODD ADDR, RST ODD ADDR FLAG
16C2	5FE9	ALDP 156	SCBYTE			HO=H1	MOVE IN THE OTHER (ODD) BYTE
16C4	5E35	ALDP 157	ENTRY			V1=HOXL	MOVE HIGH HEX DIGIT INTO V1
16C6	23FD	ALDP 158	SECHX			V1=V1+KFO	UNPACK HEX TO EBCDIC, ADD FO
16C8	0361	ALDP 159				Z=V1+K06	TEST FOR DIGIT HIGHER THAN 9
16CA	F4D0	ALDP 160		163	DONUPK	BR IF AC=0	BR IF DIGIT IS 0 THRU 9
16CC	2378	ALDP 161				V1=V1+K07	DIGIT IS A THRU F, CHANGE
16CE	23CD	ALDP 162				V1=V1+KCO	IT TO C1 THRU C6
16D0	874E	ALDP 163	DONUPK	193	TYPCHR	BAL	TYPE IT OUT
16D2	D75B	ALDP 164		168	DONE1	BR IF D1 BIT1=1	BR IF DONE ONE BYTE, LOOPING CTRL
16D4	5E3D	ALDP 165				V1=H0L	MOVE IN LOW(2ND OR 4TH)HEX DIGIT
16D6	176D	ALDP 166				D1=D1K60	INVERT LOOPING CONTROL BITS
16D8	96C6	ALDP 167		158	SECHX	BR	GO TYPE THE 2ND OR 4TH HEX DIGIT
16DA	174D	ALDP 168	DONE1			D1=D1K40	INVERT LOOPING CONTROL 1
16DC	E743	ALDP 169		156	SCBYTE	BR IF D1 BIT2=1	BR IF JUST FINISHED 1ST BYTE
16DE	22FF	ALDP 170				V0=V0+KFF	DECREMENT THE 4 HW COUNT BY 1
16E0	C4AC	ALDP 171		126	GOSPCE	BR IF ZN2	BR IF COUNTER NOT ZERO YET
16E2	3743	ALDP 172				D1=D1K04	SET 16 TH CHAR FLAG
16E4	874E	ALDP 173		193	TYPCHR	BAL	GO TO TYPE OUT THE LAST CHAR
16E6	1743	ALDP 174				D1=D1*-K04	RST THE 16TH CHAR FLAG
16E8	C36F	ALDP 175		178	NOBSTP	BR IF D1 BIT4=1	BR IF NOT BSTP NOR LOGOUT
16EA	873C	ALDP 176		182	LINEFD	BAL	FORCE A LINE FEED
16EC	8752	ALDP 177		195	NOGO	BR	DONE INST. STEP OR LOG OUT
16EE	2243	ALDP 178	NOBSTP			V0=0K04	RESTORE HW COUNTER TO 4
16F0	8744	ALDP 179		187	REDCHI	BAL	**DONE 8 BYTES, GO GET EOB OR ANY CHAR
16F2	96AE	ALDP 180		127	DECONT	BR	NOT EOB, DISPLAY 8 MORE BYTES
		ALDP 181	AEND				
073C	2315	ALDP 182	LINEFD			V1=0K10	LINE FEED SUBROUTINE
073E	3353	ALDP 183				V1=V1K05	SET V1 TO LINE FEED CHARACTER
0740	3F00	ALDP 184				SET TA K=80	ALLOW KEYBOARD RESTORE
0742	C4CE	ALDP 185		193	TYPCHR	BR IF ZN2	UNCONDITIONAL BR
		ALDP 186					STRING BREAKER
0744	2F20	ALDP 187	REDCHI			SET TA K=02	INITIALIZE PRINTER
0746	3F02	ALDP 188	READCH			SET TA K=90	SET READ LATCH, & ALDP ACTIVE
0748	1683	ALDP 189				DO=DO*-K08	SET READ CHAR SUBROUTINE FLAG
074A	C4D0	ALDP 190		194	INTVCK	BR IF ZN2	UNCONDITIONAL BR
		ALDP 191					STRING BREAKER
074C	2345	ALDP 192	SPACE			V1=0K40	SPACE SUBROUTINE, V1=BLANK
074E	2F0A	ALDP 193	TYPCHR			SET TA K=50	TYPE CHAR ROUTINE, SET WRITE LTH
0750	EEA2	ALDP 194	INTVCK	203	STRADR	BR IF TT BIT2=0	BR IF NO INTERVENTION REQUIRED
0752	1F2A	ALDP 195	NOGO			RST TA K=D2	RST RD, WT, ACTIVE & SHARE
0754	E31A	ALDP 196		199	ADDED	BR IF D1 BIT6=0	BR IF NORMAL ALDP
0756	FA9B	ALDP 197		199	ADDED	BR IF TT BIT7=1	BR IF LOGOUT LATCH ON
0758	BC02	ALDP 198		BCHK 064	ENTRY	BR	CK SUM LOGOUT DONE
071A	2400	ALDP 199	ADDED			SET MODE K=00	SET CPU MODE TO BR ON MC REG
071C	AAAF	ALDP 200		202	NOMCK N	BR IF MC=N2	BR IF LOGOUT TO BE DONE YET
071E	0610	ALDP 201				RST BC K=01	RST LOGOUT LATCH

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0720	AAA2	ALDP 202	NOMCK 0	BSWI 066	RESTRH	BR	GO TO SOFT STOP
0722	3BE9	ALDP 203	STRADR			T1=0-KEO	T=COIF
0724	76AA	ALDP 204				STH D AS,T-2	STORE D(CTRL WORD OR FLAGS)IE-1F
0726	72AA	ALDP 205				STH V AS,T-2	V(HW COUNT & WRITE DATA)IN IC-1D
0728	74AA	ALDP 206				STH G AS,T-2	STORE G (COLUMN COUNT) IN 1A-1B
072A	70A0	ALDP 207				STH U AS,T+0	STORE U(ADDR REG'S) IN 0018-0019
072C	2B2D	ALDP 208				T1=T1+K20	T=CO39
072E	7EA0	ALDP 209				STH H AS,T+0	STORE H(DATA REG'S) IN 0038-0039
0730	2440	ALDP 210				SET MODE K=04	SET TO BACK UP ZONE
0732	7872	ALDP 211				STH I DA,9E	STORE THE BAL RETURN ADDR IN K-7
0734	2400	ALDP 212	RESET			SET MODE K=00	SET TO CPU MODE & ZONE
0736	E33B	ALDP 213		215	GOBSWI	BR IF D1 BIT6=1	BR IF FROM 'BMCK' ROUTINE
0738	16C0	ALDP 214				RST BC K=80	SET SOFT STOP LATCH
073A	AAA2	ALDP 215	GOBSWI	BSWI 066	RESTRH	BR	GO TO SOFT STOP
2688	3BE9	ALDP 216	RTTYP			T1=0-KEO	T=COIF
268A	56AA	ALDP 217				RDH D AS,T-2	RESTORE D(CTRL WORD OR FLAGS)
268C	52AA	ALDP 218				RDH V AS,T-2	STORE V(HW COUNTER & WRITE DATA)
268E	54AA	ALDP 219				RDH G AS,T-2	RESTORE G(COL. COUNT)
26C0	42AF	ALDP 220				STP0=T0	DISABLE STORAGE PROTECT
26C2	C218	ALDP 221		232	RDCHRT	BR IF D0 BIT4=0	BR IF IT WAS FROM 'READCH'
26C4	D349	ALDP 222		224	BAKMOD	BR IF D1 BIT5=1	BR IF IT WAS THE 16TH CHAR
26C6	4F3F	ALDP 223				TE=V1	SEND CHAR TO BUS OUT
26C8	2440	ALDP 224	BAKMOD			SET MODE K=04	SET TO BACK UP MODE
26CA	5872	ALDP 225				RDH I DA,9E	RESTORE BAL RETURN ADDR
26CC	2404	ALDP 226				SET MODE K=20	SET BACK TO 1052 MODE
26CE	50A0	ALDP 227				RDH U AS,T+0	RESTORE U(ADDR REG'S)
26D0	2B2D	ALDP 228				T1=T1+K20	T=CO39
26D2	5EA0	ALDP 229				RDH H AS,T+0	RESTORE H(DATA REG'S)
26D4	1F28	ALDP 230				RST TA K=C2	RST READ, WRITE LATCH, SHARE REQ
26D6	128E	ALDP 231				RTN	
2698	5A4F	ALDP 232	RDCHRT			GO=TI	GET CHAR. FROM BUS IN
269A	3683	ALDP 233				DO=D0\$K08	RST 'READCH' BAL ROUTINE FLAG
269C	CAAF	ALDP 234		243	RSTLCH	BR IF TT BIT4=1	BR IF KEY BOARD CK
269E	14C5	ALDP 235				GO=GO*-KCO	ALLOW BOTH UPPER & LOWER CASES
26A0	DAB2	ALDP 236		245	NOEOBC	BR IF TT BIT5=0	BR IF NOT ALTERNATE CODE
26A2	04AB	ALDP 237				Z=GO\$K0A	MASK FOR 'CANCEL'
26A4	C4AF	ALDP 238		243	RSTLCH	BR IF Z=0	BR IF 'CANCEL'
26A6	1623	ALDP 239				DO=D0*-K02	SET INVALID CHAR. FLAG
26A8	045B	ALDP 240				Z=GO\$K05	MASK FOR EOB
26AA	C4B2	ALDP 241		245	NOEOBC	BR IF ZNZ	BR IF NOT EOB
26AC	C632	ALDP 242		245	NOEOBC	BR IF D0 BIT0=0	BR IF NOT EXPECTING EOB
26AE	1F28	ALDP 243	RSTLCH			RST TA K=C2	RST READ, WRITE LATCH, SHARE REQ
26B0	961C	ALDP 244		022	START	BR	RESTART ANOTHER PASS
26B2	B17A	ALDP 245	NOEOBC	DYPE	STTRSL	BAL	GO TO X'LATE CHAR IN H1
26B4	E201	ALDP 246		248	NOIVLD	BR IF D0 BIT6=1	BR IF INVALID CH. FLAG IS NOT ON
26B6	962A	ALDP 247	GOBAD	029	BADCHR	BR	CHAR. TYPED IS INVALID
26B8	D648	ALDP 248	NOIVLD	224	BAKMOD	BR IF D0 BIT1=0	BR IF PACKING IS NOT REQUIRED
2682	OFFD	ALDP 249				Z=H1\$KFO	MASK FOR NUMERIC
2684	E095	ALDP 250		258	DONPCK	BR IF HZ=0	BR IF CH. IS NUMERIC
2686	1FCD	ALDP 251				H1=H1\$KCO	INVERT HIGH TWO BITS
2688	EC8F	ALDP 252		255	ALPHA	BR IF HZ=0	BR IF CX
268A	1F4D	ALDP 253				H1=H1\$K40	MASK FOR 8X
268C	E0B6	ALDP 254		247	GOBAD	BR IF HZNZ	BR IF CHAR IS NOT CX NOR 8X

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
268E	F0B7	ALDP 255	ALPHA	247	GOBAD	BR IF LZ=0	BR IF CHAR IS CO OR 80 - INVALID
2690	2F9B	ALDP 256				H1=H1+K09	ADJUST
2692	E0B6	ALDP 257		247	GOBAD	BR IF HZMZ	BR IF CH. IS NOT A THRU F
2694	5F4D	ALDP 258	DONPCK			GO=H1L	DONE PACKING
2696	A6C8	ALDP 259		224	BAKMOD	BR	
		ALDP 260	ATABLE	ADDR=0044			
0044	D9C4	ALDP 261	C			CAUX'RDDA'	
		ALDP 262	AEND				
		ALDP 263	ATABLE	ADDR=0054			
0054	D9C1	ALDP 264	C			CAUX'RAHC'	
		ALDP 265	AEND				
		ALDP 266	ATABLE	ADDR=0064			
0064	40C4	ALDP 267	C			CAUX' DIL'	
		ALDP 268	AEND				
		ALDP 269	ATABLE	ADDR=0074			
0074	C1E5	ALDP 270	C			CAUX'AVNI'	
		ALDP 271	AEND				

 * CROSS REFERENCE FOR CSECT ALDP *

ALDP 022	ALDP 035	ALDP 244	DYPE 019				
ALDP 029	ALDP 044	ALDP 077	ALDP 082	ALDP 247			
ALDP 030	ALDP 116						
ALDP 032	ALDP 034	ALDP 037	ALDP 039				
ALDP 036	ALDP 040						
ALDP 041	ALDP 028						
ALDP 047	ALDP 042						
ALDP 076	ALDP 072						
ALDP 081	ALDP 078						
ALDP 082	ALDP 075						
ALDP 084	ALDP 079	ALDP 080					
ALDP 099	ALDP 095	BCHK 063					
ALDP 100	ALDP 130	BMCK 088					
ALDP 104	ALDP 125	ALDP 128					
ALDP 114	ALDP 150						
ALDP 117	ALDP 108	ALDP 114					
ALDP 126	ALDP 171						
ALDP 127	ALDP 180						
ALDP 148	ALDP 104						
ALDP 152	ALDP 118						
ALDP 156	ALDP 169						
ALDP 157	ALDP 154	BSTP 019					
ALDP 158	ALDP 167						
ALDP 163	ALDP 160						
ALDP 168	ALDP 164						
ALDP 178	ALDP 175						
ALDP 182	ALDP 024	ALDP 029	ALDP 094	ALDP 129	ALDP 176	BMCK 087	
ALDP 187	ALDP 025	ALDP 179					
ALDP 188	ALDP 047	ALDP 086	ALDP 088	ALDP 090	ALDP 092	ALDP 119	ALDP 121
ALDP 192	ALDP 085	ALDP 096	ALDP 097	ALDP 126			
ALDP 193	ALDP 033	ALDP 163	ALDP 173	ALDP 185			
ALDP 194	ALDP 190						
ALDP 195	ALDP 177						

* CROSS REFERENCE FOR CSECT ALDP *

ALDP 199	ALDP 196	ALDP 197
ALDP 202	ALDP 200	
ALDP 203	ALDP 194	
ALDP 212	BSWI 125	
ALDP 215	ALDP 213	
ALDP 216	DYPE 020	
ALDP 224	ALDP 222	ALDP 248 ALDP 259
ALDP 232	ALDP 221	
ALDP 243	ALDP 234	ALDP 238
ALDP 245	ALDP 236	ALDP 241 ALDP 242
ALDP 247	ALDP 254	ALDP 255 ALDP 257
ALDP 248	ALDP 246	
ALDP 255	ALDP 252	
ALDP 258	ALDP 250	

BCHK DESCRIPTIVE TEXT

THE CHECK SUM ROUTINE -BCHK-, IS ENTERED UPON COMPLETION OF THE RESIDENT CPU MICRODIAGNOSTIC -BDIA-. THESE ROUTINES ARE EXECUTED WHENEVER THE CSL, SYSTEM RESET, OR LOAD KEYS ARE PRESSED.

THE CHECK SUM ROUTINE PERFORMS AN EXCLUSIVE OR ON THE CONTENTS OF CONTROL STORAGE WITH THE EXCEPTION OF LOCATIONS 0002-000D AND 0280-02BF.

LOCATIONS 0002-000D CONTAIN THE HANDLOADED CHECK SUM VALUES OF THE INDIVIDUAL CORE LOADS. WHEN A CSL OPERATION IS PERFORMED, THE CHECK SUM VALUE IS SELECTED FROM THIS AREA AND PLACED IN LOCATION OEC4, WHERE IT BECOMES PART OF THE AREA THAT IS SUBJECTED TO THE EXCLUSIVE OR. THIS CHECK SUM VALUE SHOULD CAUSE THE EXCLUSIVE OR RESULT TO BE ZERO.

THE HANDLOADED AREAS ARE -

ADDRESS	CHECK SUM VALUE FOR
0002	*E60/*E61 (MODEL 25 MODE)
C004	*E62
C006	*E63
C008	*E40 (1401/1460 MODE)
000A	*E50 (1440 MODE)
C00C	*E20 (MODEL 20 MODE)

THE CHECK SUM ROUTINE IS EXECUTED IN THE FOLLOWING SEQUENCE-

- * 1 * ENTRY IS MADE FROM THE BDIA ROUTINE TO LABEL -START-
- * 2 * SET 1052 MODE AND INITIALIZE CERTAIN DIRECT ADDRESSABLE CONTROL STORAGE LOCATIONS.

- * 3 * TEST LOGOUT- *ON * A CSL OPERATION HAD BEEN PERFORMED. SET UP THE LOGOUT AREA WITH CHECK SUM INFORMATION.
 - OEC0 -- EC LEVEL
 - OEC2 -- LABEL OF CORE LOAD BEING USED.
 - OEC4 -- VALUE FROM HANDLOAD AREA.
 - OEC6 -- CORRECTION FACTOR, ALWAYS ZERO ON INITIAL LOGOUT.
 GO TO -ALDP- ROUTINE TO LOGOUT THIS INFORMATION. RETURN IS MADE TO -BCHK- AT LABEL * ENTRY *.
 - LOGOUT *OFF* ENTRY WAS MADE AFTER A SYSTEM RESET OR LOAD OPERATION HAD BEEN PERFORMED. GO TO -BCHK- LABEL * ENTRY *.
- * 4 * PERFORM EXCLUSIVE OR OF CONTROL STORAGE. SKIP LOCATIONS 0002-000D AND 0280-02BF.
- * 5 * CHECK SUM OK - BRANCH TO -BSYS- LABEL * SYSRST * CHECK SUM BAD -SET UP ADDRESS OF CORRECTION FACTOR PRIOR TO CE KEY TEST. GO TO STEP 6.
- * 6 * CE KEY ON - STORE EXCLUSIVE OR RESULT AS CORRECTION FACTOR INTO ADDRESS OEC6. BRANCH TO LOGOUT CHECK SUM AREA AND PERFORM AN EXCLUSIVE OR AGAIN. THIS SHOULD RESULT IN A GOOD CHECK SUM. CE KEY OFF- TURN ON CSL CHECK LIGHT AND ** STOP **. (HARD STOP LATCH ON)

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
			BCHK 001			T	CHECK SUM
			BCHK 002			*	
			BCHK 003			*	
			BCHK 004			*	
3024	2404		BCHK 005			START	SET 1052 MODE
3026	4632		BCHK 045				RDH D DC,8E
3028	6662		BCHK 046				STH D DC,9C
302A	FA82		BCHK 047	064	ENTRY		BR IF TT7=0
302C	21C5		BCHK 048			AGAIN	U1=0\$KCO
302E	20E3		BCHK 049				UO=0\$KOE
3030	60C4		BCHK 050				U=U+2
3032	4208		BCHK 051				RDH V CS,U+2
3034	2AC7		BCHK 052				T0=0

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
3036	53BD	BCHK 053				T1=VIL	FIND OUT WHAT
3038	C4BC	BCHK 054		056	CALARE	BR IF ZNZ	THE CSL NAME IS
303A	2B1B	BCHK 055				T1=T1+K01	ADD 1 TO LOW DIGIT OF NAME
303C	68B3	BCHK 056	CALARE			T1=T1+T1	DOUBLE THE LOW DIGIT OF NAME
303E	42A0	BCHK 057				RDH V CS,T	** USE UPDATED NAME TO READ HANDLOAD
3040	6208	BCHK 058				STH V CS,U+2	STORE HANDLOAD CHECK SUM IN OEC4
3042	0610	BCHK 059				RST BC K=01	RESET LOGOUT LATCH
3044	511B	BCHK 060				U1=U1H	U-REG=OECO
3046	2723	BCHK 061				D1=0\$K02	SET UP FLAGS
3048	20C0	BCHK 062				SET S K=0C	SET S4 AND S5
304A	968A	BCHK 063		ALDP 099	HWCNT	BR	BRANCH TO LOGOUT OECO--OEC6
3002	4252	BCHK 064	ENTRY			RDH V DC,9A	ZERO V-REGISTER
3004	4C26	BCHK 065				P=V	ZERO P-REGISTER
3006	4AC0	BCHK 066				RDH T CS,P	READ OUT CS LOCATION 0000
3008	2DEB	BCHK 067				P1=P1+K0E	SKIP HANDLOAD INFORMATION
300A	8F2C	BCHK 068	MORE	083	DOFUNC	BAL	BR TO START EXCLUSIVE OR
300C	6CC4	BCHK 069				P=P+2	INCREMENT CS ADDRESS
300E	DC0A	BCHK 070		068	MORE	BR IF P01=0	BR IF CS NOT DONE
3010	42A6	BCHK 071				V=T	MOVE LAST EX OR RESULT TO V
3012	C481	BCHK 072		082	OKSUM	BR IF Z=0	BR IF EX OR RESULT ZERO
3014	2CE3	BCHK 073				P0=0\$K0E	SET UP
3016	2DC5	BCHK 074				P1=0\$K0C	CORRECTION
3018	3D63	BCHK 075				P1=P1\$K06	FACTOR ADDRESS
301A	C99F	BCHK 076		079	OBOY	BR IF TD4=1	BRANCH IF CE KEY ON
301C	2820	BCHK 077				SET DR K=02	TURN ON CHECK SUM LIGHT AND THE
		BCHK 078	*				HARD STOP LATCH.
301E	8F2C	BCHK 079	OBOY	083	DOFUNC	BAL	GO GENERATE CORRECTION FACTOR
3020	6AC0	BCHK 080				STH T CS,P	STORE CORRECTION FACTOR IN OEC6
3022	B02C	BCHK 081		048	AGAIN	BR	GO SETUP FOR CORRECTION LOGOUT
3000	A16C	BCHK 082	OKSUM	BSYS 002	SYSRST	BR	BRANCH TO SYSTEM RESET
0F2C	CC2B	BCHK 083	DOFUNC			Z=P0\$K02	BYPASS
0F2E	C4B6	BCHK 084		088	NOTCE	BR IF ZNZ	CE
0F30	CD8D	BCHK 085				Z=P1\$K80	TRAP
0F32	C4B6	BCHK 086		088	NOTCE	BR IF ZNZ	AREA
0F34	2D4D	BCHK 087				P1=P1+K40	
0F36	42C0	BCHK 088	NOTCE			RDH V CS,P	READ FROM CONTROL STORAGE
0F38	6A21	BCHK 089				IO=TO\$V0	PERFORM THE
0F3A	6B31	BCHK 090				T1=T1\$V1	EXCLUSIVE OR
0F3C	128E	BCHK 091				RTN	
		BCHK 092	RESERVE	0280	THRU 02BE		
		BCHK 093	RESERVE	OECO	THRU OEC2		
		BCHK 094	ATABLE	ADDR=OEC4			
OEC4	C000	BCHK 095	C			XCTL'00000000'	
		BCHK 096	AEND				
		BCHK 097	ATABLE	ADDR=0388			
0388	07FF	BCHK 098	C			XCTL'07FFC48A88004E08'	
		BCHK 099	AEND				
		BCHK 100	ATABLE	ADDR=0398			
0398	G0C8	BCHK 101	C			XCTL'00C80000'	
		BCHK 102	AEND				
		BCHK 103	ATABLE	ADDR=039C			
039C	4EG8	BCHK 104	CDOIT			XCTL'4E08128E'	
		BCHK 105	AEND				

99E

o

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT
03A8	0C00	BCHK 106	ATABLE	ADDR=03A8		XCTL'0000000000000000'
		BCHK 108	C			
		BCHK 121	AEND			
03B8	2FF7	BCHK 122	ATABLE	ADDR=03B8		XCTL'2FF72F1BC4BC0F01'
		BCHK 123	C			
		BCHK 124	AEND			

 * CROSS REFERENCE FOR CSECT BCHK *

BCHK 005	BDIA 432	
BCHK 048	BCHK 081	
BCHK 056	BCHK 054	
BCHK 064	ALDP 158	BCHK 047
BCHK 068	BCHK 070	
BCHK 079	BCHK 076	
BCHK 082	BCHK 072	
BCHK 083	BCHK 068	BCHK 079
BCHK 088	BCHK 084	BCHK 086
BCHK 104	ALDP 124	ALDP 152

BCPL DESCRIPTIVE TEXT

THE BCPL ROUTINE IS NORMALLY RESIDENT IN CONTROL STORAGE, AND IS USED TO LOAD THE INITIAL RECORD OF EITHER THE CHANNEL OR NATIVE BOOTSTRAP ROUTINES.

IF THE BCPL ROUTINE HAS BEEN ALTERED, THE APPROPRIATE HANDLOAD INFORMATION MUST BE ENTERED TO BE SURE OF CORRECT CSL OPERATION.

*** SWITCH SETTINGS FOR CSL ***

SWITCHS

A,B = CC CSL FROM CHANNEL
 A,B = DD CSL FROM NATIVE 2560
 A,B = EE CSL FROM NATIVE 2540
 A,B = FF CSL FROM NATIVE 2311

SWITCHES C,D SET TO ACTUAL UNIT ADDRESS

PROCEDURE FOR PUNCHING CSL CARDS

COLUMNS

1-2 CONTAIN STARTING ADDRESS TO BE LOADED
 3 CONTAINS CODE INFORMATION
 HEX 80 = DATA IS FOR PROGRAM STORAGE
 HEX 40 = LAST CSL CARD OF DECK
 HEX 20 = DATA IS FOR AUXILIARY STORAGE
 HEX 10 = DATA IS FOR CONTROL STORAGE
 4 CONTAINS THE NUMBER OF HALFWORDS TO BE LOADED
 5-68 CONTAIN THE DATA
 69-72 OPTIONAL, CAN BE USED FOR ANY INFORMATION.
 73-77 *NNN9 WHERE N IS THE CORE LOAD ID.
 78-80 XXX 3 DIGIT NUMBER INDICATING SEQ. OF PATCHES.
 *** WARNING- DO NOT PUNCH CARDS TO LOAD INTO AUX MODULE 1,3,4

RESTRICTIONS WHEN PUNCHING CSL CARDS

COLS 1-2 ALL ADDRESSES SHOULD BE EVEN.
 CONTROL STORAGE - THESE ADDRESSES SHOULD BE IN THE RANGE OF 0000-3FFE ONLY.
 AUXILIARY STORAGE- THE 2ND HEX CHARACTER IN COLUMN 1 SHOULD BE A ZERO.
 PROGRAM STORAGE - THE ADDRESS MUST BE IN THE RANGE OF THE SYSTEM.
 COL 3 CODED INFORMATION
 COL 4 COUNT FIELD - IF A COUNT OF 0 IS INDICATED, 257 HALFWORDS WILL BE LOADED.

COLS 5-68 DATA FIELD - CAN BE ANYTHING
 COLS 69-80 SEE ABOVE.
 NOT USED BY THE CSL.
 *** WHEN LOADING A FULL DECK WITH REPLACE CARDS, THE REPLACE CARDS SHOULD GO JUST BEFORE THE END CARD.
 *** WHEN LOADING ONLY REPLACE CARDS, IT IS NECESSARY TO PUT THE CONTROL STORAGE CARDS (NORMALLY 4) FOR MODULE 01XX IN THE DECK.
 ORDER OF THE CARDS IS - BOOTSTRAP CARDS, CONTROL STORAGE CARDS FOR MODULE 01XX, REPLACE CARDS, END CARD.

** EXAMPLE OF PUNCHING AND LOADING A REPLACE CARD **

ASSUME THAT THE DEVICE ADDRESS FOR THE 1403 ON THE BURST CHANNEL NEEDED TO BE CHANGED TO -0A-.
 THE STANDARD ADDRESS FOR THE 1403 ON THE BURST CHANNEL IS -0E- AND IS LOCATED IN AUXILIARY STORAGE MODULE 0, AT ADDRESS -87-.
 THE CARD TO CHANGE THIS WOULD BE PUNCHED IN THE FOLLOWING MANNER--
 COL 1 (HEX 00) THIS ADDRESSES MODULE 0.
 COL 2 (HEX 86) ALTHOUGH THE CHANGE IS FOR ADDRESS 87, THE HEX ADDRESS REPRESENTED BY COLUMN 2 MUST BE EVEN.
 COL 3 (HEX 20) THIS INDICATES INFORMATION IS FOR AUXILIARY STORAGE.
 COL 4 (HEX 01) THIS INDICATES ONE HALFWORD TO BE LOADED.

COL 5 (HEX 0D) THIS REPRESENTS THE DEVICE ADDRESS OF THE 2540 PUNCH THAT MUST BE RELOADED BECAUSE OF THE ADDRESSING RESTRICTION IMPOSED BY COLUMN 2.
 COL 6 (HEX 0A) THIS REPRESENTS THE DEVICE ADDRESS TO INDICATE START ADDR OF PATCH.
 COL 69-72 (0086)
 COL 73-77 (*E609)
 COL 78-80 (001) INDICATES PATCH IS FOR *E60 CORE LD. INDICATES 1ST PATCH TO DECK REPLACING THE STANDARD ADDRESS FOR THE 1403 ON THE BURST CHANNEL.
 A CSL MUST BE PERFORMED TO LOAD THIS RECONFIGURATION CARD INTO THE SYSTEM. THE RECONFIGURATION CARD MUST PRECEED THE END CARD, AND MUST BE PRESENT IN THAT POSITION FOR ALL CSLS USING THAT PARTICULAR CORE LOAD DECK.
 *** EACH TIME A NEW EC LEVEL DECK IS RECIEVED, ALL RECONFIGURATION CARDS IN THE DECK BEING REPLACED MUST BE REMOVED AND INSERTED IN THE NEW DECK IN FRONT OF THE END CARD.

PUNCHED CARD CODES

HEX	PUNCHES	HEX	PUNCHES	HEX	PUNCHES	HEX	PUNCHES	HEX	PUNCHES	HEX	PUNCHES	HEX	PUNCHES
00	T-0-9-8-1	.25	0-9-5	.4A	T-8-2	.6F	0-8-7	.94	T-E-4	.B9	T-E-0-9	.DE	T-E-9-8-6
01	T-9-1	.26	0-9-6	.4B	T-8-3	.70	T-E-0	.95	T-E-5	.BA	T-E-0-8-2	.DF	T-E-9-8-7
02	T-9-2	.27	0-9-7	.4C	T-8-4	.71	T-E-0-9-1	.96	T-E-6	.BB	T-E-0-8-3	.E0	0-8-2
03	T-9-3	.28	0-9-8	.4D	T-8-5	.72	T-E-0-9-2	.97	T-E-7	.BC	T-E-0-8-4	.E1	E-0-9-1
04	T-9-4	.29	0-9-8-1	.4E	T-8-6	.73	T-E-0-9-3	.98	T-E-8	.BD	T-E-0-8-5	.E2	0-2
05	T-9-5	.2A	0-9-8-2	.4F	T-8-7	.74	T-E-0-9-4	.99	T-E-9	.BE	T-E-0-8-6	.E3	0-3
06	T-9-6	.2B	0-9-8-3	.50	T	.75	T-E-0-9-5	.9A	T-E-8-2	.BF	T-E-0-8-7	.E4	0-4
07	T-9-7	.2C	0-9-8-4	.51	T-E-9-1	.76	T-E-0-9-6	.9B	T-E-8-3	.C0	T-0	.E5	0-5
08	T-9-8	.2D	0-9-8-5	.52	T-E-9-2	.77	T-E-0-9-7	.9C	T-E-8-4	.C1	T-1	.E6	0-6
09	T-9-8-1	.2E	0-9-8-6	.53	T-E-9-3	.78	T-E-0-9-8	.9D	T-E-8-5	.C2	T-2	.E7	0-7
0A	T-9-8-2	.2F	0-9-8-7	.54	T-E-9-4	.79	8-1	.9E	T-E-8-6	.C3	T-3	.E8	0-8
0B	T-9-8-3	.30	T-E-0-9-8-1	.55	T-E-9-5	.7A	8-2	.9F	T-E-8-7	.C4	T-4	.E9	0-9
0C	T-9-8-4	.31	9-1	.56	T-E-9-6	.7B	8-3	.A0	E-0-8-1	.C5	T-5	.EA	E-0-9-8-2
0D	T-9-8-5	.32	9-2	.57	T-E-9-7	.7C	8-4	.A1	E-0-1	.C6	T-6	.EB	E-0-9-8-3
0E	T-9-8-6	.33	9-3	.58	T-E-9-8	.7D	8-5	.A2	E-0-2	.C7	T-7	.EC	E-0-9-8-4
0F	T-9-8-7	.34	9-4	.59	E-8-1	.7E	8-6	.A3	E-0-3	.C8	T-8	.ED	E-0-9-8-5
10	T-E-9-8-1	.35	9-5	.5A	E-8-2	.7F	8-7	.A4	E-0-4	.C9	T-9	.EE	E-0-9-8-6
11	E-9-1	.36	9-6	.5B	E-8-3	.80	T-0-8-1	.A5	E-0-5	.CA	T-0-9-8-2	.EF	E-0-9-8-7
12	E-9-2	.37	9-7	.5C	E-8-4	.81	T-0-1	.A6	E-0-6	.CB	T-0-9-8-3	.F0	0
13	E-9-3	.38	9-8	.5D	E-8-5	.82	T-0-2	.A7	E-0-7	.CC	T-0-9-8-4	.F1	1
14	E-9-4	.39	9-8-1	.5E	E-8-6	.83	T-0-3	.A8	E-0-8	.CD	T-0-9-8-5	.F2	2
15	E-9-5	.3A	9-8-2	.5F	E-8-7	.84	T-0-4	.A9	E-0-9	.CE	T-0-9-8-6	.F3	3
16	E-9-6	.3B	9-8-3	.60	E	.85	T-0-5	.AA	E-0-8-2	.CF	T-0-9-8-7	.F4	4
17	E-9-7	.3C	9-8-4	.61	0-1	.86	T-0-6	.AB	E-0-8-3	.D0	E-0	.F5	5
18	E-9-8	.3D	9-8-5	.62	E-0-9-2	.87	T-0-7	.AC	E-0-8-4	.D1	E-1	.F6	6
19	E-9-8-1	.3E	9-8-6	.63	E-0-9-3	.88	T-0-8	.AD	E-0-8-5	.D2	E-2	.F7	7
1A	E-9-8-2	.3F	9-8-7	.64	E-0-9-4	.89	T-0-9	.AE	E-0-8-6	.D3	E-3	.F8	8
1B	E-9-8-3	.40	NONE	.65	E-0-9-5	.8A	T-0-8-2	.AF	E-0-8-7	.D4	E-4	.F9	9
1C	E-9-8-4	.41	T-0-9-1	.66	E-0-9-6	.8B	T-0-8-3	.B0	T-E-0-8-1	.D5	E-5	.FA	T-E-0-9-8-2
1D	E-9-8-5	.42	T-0-9-2	.67	E-0-9-7	.8C	T-0-8-4	.B1	T-E-0-1	.D6	E-6	.FB	T-E-0-9-8-3
1E	E-9-8-6	.43	T-0-9-3	.68	E-0-9-8	.8D	T-0-8-5	.B2	T-E-0-2	.D7	E-7	.FC	T-E-0-9-8-4
1F	E-9-8-7	.44	T-0-9-4	.69	0-8-1	.8E	T-0-8-6	.B3	T-E-0-3	.D8	E-8	.FD	T-E-0-9-8-5
20	E-0-9-8-1	.45	T-0-9-5	.6A	T-E	.8F	T-0-8-7	.B4	T-E-0-4	.D9	E-9	.FE	T-E-0-9-8-6
21	0-9-1	.46	T-0-9-6	.6B	0-8-3	.90	T-E-8-1	.B5	T-E-0-5	.DA	T-E-9-8-2	.FF	T-E-0-9-8-7
22	0-9-2	.47	T-0-9-7	.6C	0-8-4	.91	T-E-1	.B6	T-E-0-6	.DB	T-E-9-8-3		
23	0-9-3	.48	T-0-9-8	.6D	0-8-5	.92	T-E-2	.B7	T-E-0-7	.DC	T-E-9-8-4		
24	0-9-4	.49	T-8-1	.6E	0-8-6	.93	T-E-3	.B8	T-E-0-8	.DD	T-E-9-8-5		

*** HANDLOAD ROUTINE FOR NATIVE 2540 ***

ADDR	WORD	STATEMENT	COMMENT
0010	3210	SET MMSK K=81	BLOCK TRAPS
0012	2610	SET BC K=01	SET LOGOUT LATCH
0014	2C07	PO=0	ZERO OUT SWITCH
0016	2413	GO=0\$K01	START SETUP OF ADDR 0100
0018	8062	BR	BRANCH TO LOCATION 0062
0062	2406	SET MODE K=30	SET 2540 MODE
0064	2617	DO=0\$K11	BUILD AUXILIARY
0066	2783	D1=0\$K08	STORAGE ADDRESS
0068	3775	D1=D1\$K70	FOR COMPARING
006A	2507	G1=0	FINISH SETUP OF ADDR 0100
006C	4066	U=D	SAVE COMPARE ADDRESS
006E	5EEF	HO=RPS	READ 2540 STATUS
0070	DE6E	BR IF H01=0	BACK ONE WORD IF NO DATA
0072	5B3F	V1=RP1	GET READ BRUSH 1 DATA
0074	6348	STB V1 CS,G+1	PUT DATA IN CONTROL STOR
0076	736A	STB V1 AS,D-1	PUT DATA IN COMP TABLE
0078	F0EE	BR IF LZNZ	GO WAIT FOR MORE DATA
007A	161B	DO=DO\$K01	INVERT FIRST TIME SWITCH
007C	F0E7	BR IF LZ=0	BRANCH BACK IF FIRST TIME
007E	8100	BR	BRANCH TO BOOTSTRAP

*** HANDLOAD ROUTINE FOR CHANNEL ***

ADDR	WORD	STATEMENT	COMMENT
0010	3210	SET MMSK K=81	BLOCK TRAPS
0012	2610	SET BC K=01	SET LOGOUT LATCH
0014	2C07	PO=0	ZERO OUT SWITCH
0016	2413	GO=0\$K01	START SETUP OF ADDR 0100
0018	51AF	TO=SWCD	SWCD EQUAL DEVICE ADDR.
001A	802C	BR	BRANCH TO LOCATION 002C
002C	2486	SET MODE K=38	SET CHANNEL MODE,CPU ZONE
002E	2507	G1=0	FINISH SETUP OF ADDR 0100
0030	2B08	SET GA K=40	SET SERVICE OUT
0032	C9B3	BR IF GT4=1	BRANCH ON OP IN
0034	4FAF	GB/OUT=TO	SEND DEVICE ADDRESS OUT
0036	2B04	SET GA K=20	RAISE ADDRESS OUT
0038	2B44	SET GA K=24	AND SELECT OUT
003A	C9BA	BR IF GT4=0	WAIT FOR OP IN
003C	2B40	SET GA K=04	RESET ADDRESS OUT
003E	CDBE	BR IF GT0=0	WAIT FOR ADDRESS IN
0040	2B23	T1=0\$K02	BUILD READ COMMAND
0042	4FBF	GB/OUT=T1	SEND OUT READ COMMAND
0044	2B42	SET GA K=14	RAISE COMMAND OUT
0046	FDC6	BR IF GT3=0	WAIT HERE FOR STATUS
0048	5FBF	T1=GB/IN	READ STATUS
004A	C4CA	BR IF ZNZ	LOOP HERE IF INVALID STAT
004C	2B48	SET GA K=44	SET SERVICE OUT
004E	FDCF	BR IF GT3=1	WAIT FOR
0050	EDCE	BR IF GT2=0	DATA
0052	5FFF	H1=GB/IN	GET DATA BYTE
0054	F05B	BR IF G07=1	BR IF BOOTSTRAP READING
0056	7F48	STB H1 AS,G+1	NOT BOOT INFO,STOR IN AUX
0058	F05C	BR IF G07=0	UNCONDITIONAL BRANCH
005A	6F48	STB H1 CS,G+1	PUT BOOT IN CNTRL STORAGE
005C	055D	Z=G1\$K50	CHECK IF ALL DATA IN,
005E	C4CC	BR IF ZNZ	IF NOT, GET MORE.
0060	8100	BR	BRANCH TO BOOTSTRAP

*** HANDLOAD ROUTINE FOR 2311 ***

ADDR	WORD	STATEMENT	COMMENT
0010	3210	SET MMSK K=81	BLOCK TRAPS
0012	2610	SET BC K=01	SET LOGOUT LATCH
0014	2CC7	PO=0	ZERO OUT SWITCH
0016	8076	BR	BRANCH TO LOCATION 0076
0076	2490	SET MODE K=09	SET 2311 MODE
0078	2813	IO=0\$K01	SET I-REG
007A	2907	I1=0	TO 0100
007C	4A86	T=I	MOVE 0100 TO T-REGISTER
007E	2F15	H1=0\$K10	
0080	2EX5	HO=0\$KX0	SET UP MODULE SELECT * NOTE *
0082	2D23	P1=0\$K02	SET UP RETURN BIT (BIT 6)
0084	4D9F	FFD=I1	SEND ZERO TO FILE FLAGS OUT
0086	0E08	RST FIB K=40	ISSUE INITIAL RESET
0088	0E04	RST FIB K=20	ISSUE COLD-START RESET
008A	49EF	MS=HO	LOAD MODULE SELECT REGISTER
008C	4EDF	FBO=P1	SEND RETURN TO FILE BUS OUT
008E	4BFF	TGRO=H1	MOVE CONTROL BIT TO TAG REG
0090	2E63	HO=0\$K06	SET UP FILE OP, COUNT
0092	3E15	HO=HO\$K10	OF ONE, DATA READ
0094	4FEF	FOP=HO	MOVE OP TO FILE OP REGISTER
0096	2E43	HO=C\$K04	SET UP SELECT HEAD
0098	E598	BR IF DAS12=0	BR TO ITSELF IF NO GATED ATT.
009A	4EEF	FBO=HO	MOVE 04 TO FILE BUS OUT
009C	4BFF	TGRO=H1	MOVE CNTRL BIT TO TAG REG OUT
009E	3D00	SET FIA K=80	SET GO LATCH
00A0	E5A1	BR IF DAS12=1	BR TO ITSELF IF GATED ATTEN.
00A2	1D00	RST FIA K=80	RESET TRAP LATCH
00A4	50EF	HO=TGRI	GET TAG REGISTER IN
00A6	CA25	BR IF HO4=1	BR BACK ONE WORD IF NO TRAP
00A8	C1A9	BR IF DAS14=1	LOOP ON ITSELF IF ERROR
00AA	810A	BR TO 010A	BR TO BOOTSTRAP

* NOTE * THE X IN THE CONTROL WORD HAS THE FOLLOWING SIGNIFICANCE-

X=8 SELECT DRIVE NUMBER 0
 X=4 SELECT DRIVE NUMBER 1
 X=2 SELECT DRIVE NUMBER 2
 X=1 SELECT DRIVE NUMBER 3

*** HANDLOAD ROUTINE FOR 2560 ***

ADDR	WORD	STATEMENT	COMMENT
0010	3210	SET MMSK K=81	BLOCK ALL TRAPS
0012	2610	SET BC K=01	SET LOGOUT LATCH
0014	2C07	PO=0	ZERO HANDLOAD FLAG REGISTER
0016	2413	GO=0\$K01	BUILD HIGH HALF CS ADDR. 0100
0018	80AC	BR	BR TO 00AC
00AC	240E	SET MODE K=70	PUT IN MOD/20, 2560 MODE
00AE	2F04	SET MFA K=20	SELECT SEC FEED. (NOTE 2)
00B0	2507	G1=0	LOW HALF CS ADDR. G=0100
00B2	5FDF	P1=MFD8	
00B4	DAB8	BR IF MFT5=0	BR TO 00B8 IF NPRO REQ
00B6	ED33	BR IF P12=1	BR IF NOT READY
00B8	2F10	SET MFA K=01	SET READ EX.
00BA	DAC6	BR IF MFT5=0	CHECK NPRO
00BC	CAB9	BR IF MFT4=1	BR ON NO DATA AVAIL
00BE	58FF	H1=MFR1	READ 1/2 BYTE
00C0	5AEF	HO=MFR2	READ THE OTHER 1/2
00C2	4EF3	H1=HOXH+H1L	PUT TWO 1/2 BYTES TOGETHER
00C4	6F48	STB H1 CS,G+1	STORE THEM
00C6	2B14	SET MFC K=21	RST NPRO, AND RD. EX.
00C8	055D	Z=G1\$K50	CHECK FOR 80 BYTES
00CA	C4B4	BR IF ZNZ	IF NOT 80, GO TO 00B4
00CC	8100	BR	BR TO BOOTSTRAP ADDR 0100

NOTE 2 - IN ORDER TO USE THE PRIMARY FEED, REPLACE THIS WORD WITH 2F80. THE START KEY MUST BE PRESSED AT THE END TO COMPLETE THE CSL (LAST CARD).

 *
 * FOR BOOTSTRAP *
 * INFORMATION, REFER *
 * TO THE AKXXX LOGIC *
 * PAGES. *

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BCPL 001	T	CSL		RESIDENT ROUTINE	
		BCPL 002	ATABLE	ADDR=0000			
0000	8000	BCPL 003	LOCKON	003	LOCKON	BR	
0002	0BAD	BCPL 004	C			XCTL'0BAD0BAD0BAD0BAD0BAD0'	
000E	8CAC	BCPL 005	N2560	087	RD2560	BR	
		BCPL 006	*			THE FOLLOWING WORD	
		BCPL 007	*			IS THE FIRST WORD	
		BCPL 008	*			OF THE CSL TRAP	
0010	3210	BCPL 009				SET MMSK K=81	BLOCK ALL TRAPS
0012	2610	BCPL 010				SET BC K=01	SET LOG OUT LATCH
0014	50CF	BCPL 011				PO=SWAB	READ
0016	51AF	BCPL 012				TO=SWCD	SWITCHES
0018	2413	BCPL 013				GO=0\$K01	SET UP C S ADDRESS
001A	0CCF	BCPL 014				Z=PO\$KCC	CHECK FOR
001C	C4AD	BCPL 015		023	CHANEL	BR IF Z=0	CHANNEL
001E	0CDF	BCPL 016				Z=PO\$KDD	CHECK FOR
0020	C48F	BCPL 017		005	N2560	BR IF Z=0	2560
0022	0CEF	BCPL 018				Z=PO\$KEE	CHECK FOR
0024	C4E3	BCPL 019		050	N2540	BR IF Z=0	2540
0026	0CFF	BCPL 020				Z=PO\$KFF	CHECK FOR
0028	C4A8	BCPL 021	SWIERR	021	SWIERR	BR IF ZNZ	2311
002A	80DE	BCPL 022		106	FILE	BR	
002C	2486	BCPL 023	CHANEL			SET MODE K=38	CHAN PUT IN CHANNEL MODE
002E	2507	BCPL 024	AGAIN			G1=0	CHAN CS ADDRESS
0030	28C8	BCPL 025				SET GA K=40	CHAN
0032	C9B3	BCPL 026	OPIN	026	OPIN	BR IF GT4=1	CHAN BR ON OP IN
0034	4FAF	BCPL 027				GB/OUT=T0	CHAN SEND OUT DEVICE ADDRESS
0036	28C4	BCPL 028				SET GA K=20	CHAN RAISE ADDR.
0038	2B44	BCPL 029				SET GA K=24	CHAN AND SEL OUT
003A	C9BA	BCPL 030	OPINOT	030	OPINOT	BR IF GT4=0	CHAN WAIT FOR OP IN
003C	2B40	BCPL 031				SET GA K=04	CHAN RESET ADDR. OUT
003E	0DBE	BCPL 032	ADDIN	032	ADDIN	BR IF GT0=0	CHAN WAIT FOR ADDR IN
0040	2B23	BCPL 033				T1=0\$K02	CHAN BUILD READ CMND
0042	4FBF	BCPL 034				GB/OUT=T1	CHAN SEND OUT THE CMND
0044	2B42	BCPL 035				SET GA K=14	CHAN RAISE COMMAND OUT
0046	FDC6	BCPL 036	STATUS	036	STATUS	BR IF GT3=0	CHAN WAIT FOR STATUS
0048	5FBF	BCPL 037				T1=GB/IN	CHAN READ STATUS
004A	C4CA	BCPL 038	ERRORA	038	ERRORA	BR IF ZNZ	CHAN LOOP INVALID INITIAL STATS
004C	2B48	BCPL 039	STATC			SET GA K=44	CHAN SET SERVICE OUT
004E	FDCF	BCPL 040	STATB	040	STATB	BR IF GT3=1	CHAN STATUS LOOP
0050	EDCE	BCPL 041		040	STATB	BR IF GT2=0	CHAN SERVICE LOOP
0052	5FFF	BCPL 042				H1=GB/IN	CHAN GET DATA
0054	F05B	BCPL 043		046	FIRSTC	BR IF G07=1	CHAN FIRST TIME BOOTSTRAP
0056	7F48	BCPL 044				STB H1 AS,G+1	CHAN NOT 1ST PUT IN AUX
0058	F05C	BCPL 045		047	DECREM	BR IF G07=0	CHAN UNCONDIT BRANCH
005A	6F48	BCPL 046	FIRSTC			STB H1 CS,G+1	CHAN 1ST TIME PUT IN C S
005C	055D	BCPL 047	DECREM			Z=G1\$K50	CHAN CHECK IF ALL
005E	C4CC	BCPL 048		039	STATC	BR IF ZNZ	CHAN BYTES XFERED
0060	8100	BCPL 049	C			XCTL'8100'	CHAN GO TO BOOTSTRAP
0062	2406	BCPL 050	N2540			SET MODE K=30	2540 PUT IN 2540 MODE
0064	2617	BCPL 051				DO=0\$K11	2540 BUILD COMPARE
0066	2783	BCPL 052	CYCLE			D1=0\$K08	2540 ADDRESS WHERE CARD
0068	3775	BCPL 053				D1=D1\$K70	2540 IMAGE IS STORED

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
006A	25C7	BCPL 054				G1=0	2540 ZERO REG FOR CONTROL ADDR.
006C	4066	BCPL 055				U=D	2540 SAVE COMPARE ADDRESS
006E	5EEF	BCPL 056	DATAIN			H0=RPS	2540 WAIT FOR
0070	DE6E	BCPL 057		056	DATAIN	BR IF H01=0	2540 DATA AVAILABLE
0072	5B3F	BCPL 058				V1=RP1	2540 READ 1ST BRUSH DATA
0074	6348	BCPL 059				STB V1 CS,G+1	2540 -STORE IN CONTROL STORE
0076	736A	BCPL 060				STB V1 AS,D-1	2540 -STORE IN AUX COMPARE TBLE
0078	F0EE	BCPL 061		056	DATAIN	BR IF LZNZ	2540 STAY IN DATA LOOP FOR 1 CD
007A	161B	BCPL 062				DO=D0K01	2540 INVERT FLAG
007C	F0E7	BCPL 063		052	CYCLE	BR IF LZ=0	2540 CHECK FOR 2ND TRAP CYCLE
007E	8100	BCPL 064	C			XCTL'8100'	2540 GO TO BOOTSTRAP
0080	6EE3	BCPL 065	MOD	0		H0=H0+H0	2311 BUILD
0082	6EE3	BCPL 066	MOD	1		H0=H0+H0	2311 MODULE
0084	6EE3	BCPL 067	MOD	2		H0=H0+H0	2311 NUMBER
0086	0E08	BCPL 068	MOD	3		RST FIB K=40	2311 COLD START RESET
0088	0E04	BCPL 069				RST FIB K=20	2311 COLD START RESET
008A	49EF	BCPL 070				MS=H0	2311 SELECT MODULE
008C	4EDF	BCPL 071				FBO=P1	2311 SET RETURN TO ZERO P1=02
008E	4BFF	BCPL 072				TGRO=H1	2311 SET CONTROL H1=10
0090	2E63	BCPL 073				H0=0\$K06	2311 SET READ
0092	3E15	BCPL 074				H0=H0\$K10	2311
0094	4FEF	BCPL 075				FDP=H0	2311 DATA OP
0096	2E43	BCPL 076				H0=0\$K04	2311 START HEAD SELECT
0098	E598	BCPL 077	WAITFL	077	WAITFL	BR IF DAS12=0	2311 WAIT FOR END
009A	4EEF	BCPL 078				FBO=H0	2311 HEAD
009C	4BFF	BCPL 079				TGRO=H1	2311 SELECT
009E	3D00	BCPL 080				SET FIA K=80	2311 GO
00A0	E5A1	BCPL 081	WAIENA	081	WAIENA	BR IF DAS12=1	2311 WAIT FOR
00A2	1D00	BCPL 082				RST FIA K=80	2311 RESET TRAP REQ
00A4	50EF	BCPL 083	WAIENB			H0=TGRI	2311 WAIT FOR FILE
00A6	CA25	BCPL 084		083	WAIENB	BR IF H04=1	2311 TRAP REQUEST
00A8	C1A9	BCPL 085	ER	085	ER	BR IF DAS14=1	2311 ERROR LOOP
00AA	810A	BCPL 086	C			XCTL'810A'	2311 GO TO BOOTSTRAP
00AC	240E	BCPL 087	RD2560			SET MODE K=70	2560 PUT IN 20,2560 MODE
00AE	25C7	BCPL 088				G1=0	2560
00B0	2F04	BCPL 089				SET MFA K=20	2560 SELECT SEC FEED
00B2	5FDF	BCPL 090	K5RDY			P1=MFD8	2560 CALL IN FOR RDY CHK
00B4	DAB8	BCPL 091	K6NPO	093	FEED	BR IF MFT5=0	2560 BR IF NPRO REQ
00B6	ED33	BCPL 092		090	K5RDY	BR IF P12=1	2560 BR IF NOT RDY
00B8	2F10	BCPL 093	FEED			SET MFA K=01	2560 SET READ EXECUT
00BA	DAC6	BCPL 094		100	NDIT	BR IF MFT5=0	2560 BRANCH IF NPRO
00BC	CAB9	BCPL 095		093	FEED	BR IF MFT4=1	2560 BRANCH IF NO DATA
00BE	58FF	BCPL 096				H1=MFR1	READ 1 / 2 BYTE
00C0	5AEF	BCPL 097				H0=MFR2	READ OTHER HALF
00C2	4EF3	BCPL 098				H1=H0XH+H1L	PUT THEM TOGETHER
00C4	6F48	BCPL 099				STB H1 CS,G+1	2560 STORE DATA IN C S
00C6	2B14	BCPL 100	NDIT			SET MFC K=21	2560 RST RD EX AND NPRO
00C8	055D	BCPL 101				Z=G1K50	2560
00CA	C4B4	BCPL 102		091	K6NPO	BR IF ZNZ	2560 NOT DONE BRANCH
00CC	8100	BCPL 103	C			XCTL'8100'	2560 GO TO BOOTSTRAP
00CE	FFFF	BCPL 104	C			XCTL'FFFFFFFFFFFFFFFFFFFFFFFF'	
00DC	FFFF	BCPL 105	C			XCTL'FFFF'	
00DE	2B07	BCPL 106	FILE			T1=0	2311 CORRECT PARITY

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
00E0	6A02	BCPL 107				STH T DC,88	2311 SAVE MODULE
00E2	2490	BCPL 108				SET MODE K=09	2311 SET FILE MODE ZONE
00E4	2813	BCPL 109				I0=0\$K01	2311 BUILD COUNT AND
00E6	2907	BCPL 110				I1=0	2311 CONTROL STORAGE
00E8	4A86	BCPL 111				T=I	2311 ADDRESS
00EA	4C02	BCPL 112				RDH P DC,88	2311 RETRIEVE MODULE
00EC	2F15	BCPL 113				H1=0\$K10	2311 CONSTANTS FOR
00EE	5FE9	BCPL 114				H0=H1	2311 SELECTION
00F0	2D23	BCPL 115				P1=0\$K02	2311
00F2	4D9F	BCPL 116				FF0=I1	2311
00F4	8811	BCPL 117		065	MOD N	N=PO BITS67	2311
00F6	54E8	BCPL 118	OVRLAY			RDH G AS,H+2	END OVERLAY AUX 1000
00F8	6488	BCPL 119				STH G CS,I+2	END INTO CONTROL
00FA	FA76	BCPL 120		118	OVRLAY	BR IF H07=0	END 0100
00FC	06C8	BCPL 121				RST BC K=40	END RESET CSL LIGHT
00FE	88FE	BCPL 122	C			XCTL'88FE'	END BRANCH TO 08FE
		BCPL 123	AEND				

 * CROSS REFERENCE FOR CSECT BCPL *

BCPL 003	BCPL 003	
BCPL 005	BCPL 017	
BCPL 021	BCPL 021	
BCPL 023	BCPL 015	
BCPL 026	BCPL 026	
BCPL 030	BCPL 030	
BCPL 032	BCPL 032	
BCPL 036	BCPL 036	
BCPL 038	BCPL 038	
BCPL 039	BCPL 048	
BCPL 040	BCPL 040	BCPL 041
BCPL 046	BCPL 043	
BCPL 047	BCPL 045	
BCPL 050	BCPL 019	
BCPL 052	BCPL 063	
BCPL 056	BCPL 057	BCPL 061
BCPL 065	BCPL 117	
BCPL 077	BCPL 077	
BCPL 081	BCPL 081	
BCPL 083	BCPL 084	
BCPL 085	BCPL 085	
BCPL 087	BCPL 005	
BCPL 090	BCPL 092	
BCPL 091	BCPL 102	
BCPL 093	BCPL 091	BCPL 095
BCPL 100	BCPL 094	
BCPL 106	BCPL 022	
BCPL 118	BCPL 120	

BDIA DESCRIPTIVE TEXT

THE RESIDENT MICRODIAGNOSTIC -BDIA- IS ENTERED WHEN

1. THE SYSTEM RESET KEY IS RELEASED
(TRAP TO ADDRESS 0240)
2. THE LOAD KEY IS RELEASED
(TRAP TO ADDRESS 0240)
3. THE CONTROL STORAGE LOAD ROUTINE -BCPL- IS
FINISHED LOADING A CSL DECK.
(BRANCH TO LABEL *VERSON*)

THE -BDIA- ROUTINE TESTS THE CPU HARDWARE NEEDED TO PERFORM A CONTROL STORAGE LOAD OPERATION. NO I/O DEVICE OR ATTACHMENT CIRCUITS ARE TESTED.

THE TESTS PERFORMED BY THE -BDIA- ROUTINE ARE -
BRANCHING TEST
ALU TEST
MODE REGISTER SET/RST TEST
STORAGE TEST
X LINE ADDRESSING TEST
LOCAL STORAGE SET/RST TEST
ALU ERROR DETECTION TEST
STORAGE DATA, CONTROL WORD, STORAGE ADDRESS TEST
A AND B REGISTER PARITY DETECTION TEST

ERRORS ARE INDICATED BY

1. A ONE WORD BRANCH LOOP
2. A STOP WORD
3. A BRANCH TEST STOP FAILURE

A ONE WORD BRANCH LOOP IS IDENTIFIED BY THE SYSTEM LIGHT ON AND THE MANUAL LIGHT OFF. THE CLOCK WILL RUN BUT THE MICROPROGRAM DOES NOT PROGRESS.

A STOP WORD (MOVE/ARITH-WORD TYPE 3) CAUSES THE CLOCK TO STOP, THE CLOCK STOP LIGHT IS ON, THE SYSTEM LIGHT IS OFF, THE MANUAL LIGHT IS ON, AND THE ADDRESS DISPLAYED IS THE ADDRESS OF THE CONTROL WORD FOLLOWING THE STOP WORD.

A BRANCH TEST STOP FAILURE IS CAUSED BY A BRANCH ON CONDITION WORD OR RETURN WORD FAILING TO REACH A WORD THAT SETS THE DR-REGISTER. DR BIT 7 IS SET PRIOR TO EACH OF THE BRANCH TESTS, AND THE BRANCH OR RETURN MUST POINT TO A SET OF THE DR-REGISTER. THE FAILURE IS INDICATED BY THE CLOCK OFF, THE CLOCK STOP LIGHT IS ON, SYSTEM LIGHT OFF, MANUAL LIGHT ON, AND, THE ADDRESS OF THE NEXT SEQUENTIAL WORD FOLLOWING THE WORD REACHED IN ERROR, DISPLAYED IN THE CONSOLE LIGHTS.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BDIA 001	T		BDIA	BASIC DIAGNOSTIC	
		BDIA 002	ATABLE	ADDR=08FE			
08FE	8240	BDIA 008	VERSON	018	START	BR	START RESIDENT DIAG.
		BDIA 012	AEND				
		BDIA 013	ATABLE	ADDR=0240			
		BDIA 014	*				DEPRESSION OF THE SYSTEM RESET
		BDIA 015	*				BUTTON OR LOAD BUTTON WILL CAUSE
		BDIA 016	*				A HARDWARE TRAP TO ADDRESS 0240.
		BDIA 017	*				
0240	2810	BDIA 018	START			SET DR K=01	SET DR-7. DR 7 IS USED IN
		BDIA 019	*				CHECKING THE BRANCHING FUNCTIONS
		BDIA 020	*				OF THE MOD 25. A DIAG BR LATCH
		BDIA 021	*				IS SET EVERYTIME A MOD 25 BRANCH
		BDIA 022	*				OR RETURN FUNCTION IS EXECUTED
		BDIA 023	*				AND DR 7 IS ON. IF THE NEXT
		BDIA 024	*				MICROINSTRUCTION AFTER THE BR
		BDIA 025	*				OR RETURN DOES NOT ISSUE A SET
		BDIA 026	*				TO THE DR REG, A HARD STOP WILL
		BDIA 027	*				OCCUR AT THE COMPLETION OF THAT
		BDIA 028	*				MACHINE CYCLE.
		BDIA 029	*				

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0242	3210	BDIA 030				SET MMSK K=81	SET SYSTEM RESET PRIORITY BIT(MMSK-8).
		BDIA 031	*				
		BDIA 032	*				
		BDIA 033	*				
		BDIA 034	*				THE FIRST WORD OF THE TRAP SETS THE DIAG BR LATCH TO INSURE A MACHINE STOP IF AN ERRONEOUS BRANCH STATEMENT IS EXECUTED
		BDIA 035	*				THE SECOND WORD OF THE TRAP SETS MMSK 8 TO PREVENT FURTHER TRAPS FOR THE SAME SYST RST/LOAD TRAP REQUEST. AS A RESULT OF THE DELAYED SETTING OF MMSK 8, THE TRAP WILL OCCUR TWICE FOR EACH SYSTEM RST/LOAD TRAP REQUEST. THEREFORE, THE FIRST TWO WORDS WILL BE EXECUTED TWICE BEFORE THE FOLLOWING WORD IS EXECUTED.
		BDIA 036	*				
		BDIA 037	*				
		BDIA 038	*				
		BDIA 039	*				
		BDIA 040	*				
		BDIA 041	*				
		BDIA 042	*				
		BDIA 043	*				
		BDIA 044	*				
		BDIA 045	*				
		BDIA 046	*				
		BDIA 047	*				
0244	2400	BDIA 048				SET MODE K=00	
		BDIA 049	*				STORAGE ZONE 4.
		BDIA 050	*				*****
		BDIA 051	*			BRANCHING TEST.	
		BDIA 052	*				*****
0246	F8CB	BDIA 053		056	BCDR7	BR IF DR BIT7=1	BR TO 024A IF DR-7 IS ON
0248	B7C9	BDIA 054		122	DR7OFF N	N=BAH	DR 7 SHOULD BE ON. BRANCH ON BA HIGH FOR BETTER RESOLUTION. SET DR-7 TO CONTINUE BR TESTS.
		BDIA 055	*				ATTEMPT TO SET G1 REG = 00001111 IF A STOP OCCURS AT THE ADDRESS OF THE NEXT WORD, G1 WAS SET WRONG OR THE BRANCH TEST FAILED. G10=0 SHOULD NOT BRANCH.
024A	2810	BDIA 056	BCDR7			SET DR K=01	SET DR-7 TO CONTINUE BR TESTS.
024C	25F3	BDIA 057				G1=0\$KOF	ATTEMPT TO SET G1 REG = 00001111 IF A STOP OCCURS AT THE ADDRESS OF THE NEXT WORD, G1 WAS SET WRONG OR THE BRANCH TEST FAILED. G10=0 SHOULD NOT BRANCH.
		BDIA 058	*				IF A STOP OCCURS AT THE ADDRESS OF THE NEXT WORD, G1 WAS SET WRONG OR THE BRANCH TEST FAILED. G10=0 SHOULD NOT BRANCH.
		BDIA 059	*				IF A STOP OCCURS AT THE ADDRESS OF THE NEXT WORD, G1 WAS SET WRONG OR THE BRANCH TEST FAILED. G10=0 SHOULD NOT BRANCH.
		BDIA 060	*				IF A STOP OCCURS AT THE ADDRESS OF THE NEXT WORD, G1 WAS SET WRONG OR THE BRANCH TEST FAILED. G10=0 SHOULD NOT BRANCH.
024E	C54F	BDIA 061	BCG10N	061	BCG10N	BR IF G1 BIT0=1	SET DR-7 TO CONTINUE BR TESTS.
0250	2810	BDIA 062				SET DR K=01	IF A STOP OCCURS AT THE ADDRESS OF THE NEXT WORD, G1 WAS SET WRONG OR THE BRANCH TEST FAILED. G10=0 SHOULD NOT BRANCH.
		BDIA 063	*				IF A STOP OCCURS AT THE ADDRESS OF THE NEXT WORD, G1 WAS SET WRONG OR THE BRANCH TEST FAILED. G10=0 SHOULD NOT BRANCH.
		BDIA 064	*				IF A STOP OCCURS AT THE ADDRESS OF THE NEXT WORD, G1 WAS SET WRONG OR THE BRANCH TEST FAILED. G10=0 SHOULD NOT BRANCH.
		BDIA 065	*				IF A STOP OCCURS AT THE ADDRESS OF THE NEXT WORD, G1 WAS SET WRONG OR THE BRANCH TEST FAILED. G10=0 SHOULD NOT BRANCH.
0252	D553	BDIA 066	BCG11N	066	BCG11N	BR IF G1 BIT1=1	SET DR-7 TO CONTINUE BR TESTS.
0254	2810	BDIA 067				SET DR K=01	IF A STOP OCCURS AT THE NEXT SEQUENTIAL WORD FOLLOWING THIS BRANCH, G1 WAS SET WRONG OR THE BRANCH TEST FAILED.
		BDIA 068	*				IF A STOP OCCURS AT THE NEXT SEQUENTIAL WORD FOLLOWING THIS BRANCH, G1 WAS SET WRONG OR THE BRANCH TEST FAILED.
		BDIA 069	*				IF A STOP OCCURS AT THE NEXT SEQUENTIAL WORD FOLLOWING THIS BRANCH, G1 WAS SET WRONG OR THE BRANCH TEST FAILED.
		BDIA 070	*				IF A STOP OCCURS AT THE NEXT SEQUENTIAL WORD FOLLOWING THIS BRANCH, G1 WAS SET WRONG OR THE BRANCH TEST FAILED.
		BDIA 071	*				IF A STOP OCCURS AT THE NEXT SEQUENTIAL WORD FOLLOWING THIS BRANCH, G1 WAS SET WRONG OR THE BRANCH TEST FAILED.
0256	E53C	BDIA 072		075	BCG12Y	BR IF G1 BIT2=0	G12=0 SHOULD BRANCH TO ADR 023C
		BDIA 073	AEND				
		BDIA 074	ATABLE ADDR=023C				
023C	2810	BDIA 075	BCG12Y			SET DR K=01	SET DR-7 TO CONTINUE BR TESTS.
		BDIA 076	*				IF A STOP OCCURS AT THE NEXT SEQUENTIAL WORD FOLLOWING THIS BRANCH, G1 WAS SET WRONG OR THE BRANCH TEST FAILED.
		BDIA 077	*				IF A STOP OCCURS AT THE NEXT SEQUENTIAL WORD FOLLOWING THIS BRANCH, G1 WAS SET WRONG OR THE BRANCH TEST FAILED.
		BDIA 078	*				IF A STOP OCCURS AT THE NEXT SEQUENTIAL WORD FOLLOWING THIS BRANCH, G1 WAS SET WRONG OR THE BRANCH TEST FAILED.
		BDIA 079	*				IF A STOP OCCURS AT THE NEXT SEQUENTIAL WORD FOLLOWING THIS BRANCH, G1 WAS SET WRONG OR THE BRANCH TEST FAILED.
023E	F56C	BDIA 080		083	BCG13Y	BR IF G1 BIT3=0	G13=0 SHOULD BRANCH TO ADR 026C
		BDIA 081	AEND				
		BDIA 082	ATABLE ADDR=026C				

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
026C	2810	BDIA 083	BCG13Y			SET DR K=01	SET DR-7 TO CONTINUE BR TESTS.
026E	9FFC	BDIA 084		087	UB1FFC	BR	UNCONDITIONAL BRANCH TO ADR 1FFC
		BDIA 085	AEND				
		BDIA 086	ATABLE	ADDR=1FFC			
1FFC	2810	BDIA C87	UB1FFC			SET DR K=01	SET DR-7 TO CONTINUE BR TESTS.
1FFE	D57F	BDIA 088	HCTRL			BR IF G1 BIT1=1	NO BRANCH
		BDIA 089	AEND				
		BDIA 090	ATABLE	ADDR=2000			
2000	2810	BDIA 091				SET DR K=01	SET DR-7 TO CONTINUE BR TESTS.
2002	25F5	BDIA 092				G1=0\$KFO	ATTEMPT TO SET G1 REG = 11110000
		BDIA 093	*				IF A STOP OCCURS AT THE ADDRESS
		BDIA 094	*				OF THE NEXT WORD, G1 WAS SET
		BDIA 095	*				WRONG OR THE BRANCH TEST FAILED.
2004	E504	BDIA 096	BCG12N	096	BCG12N	BR IF G1 BIT2=0	G12=1 SHOULD NOT BRANCH
2006	2810	BDIA 097				SET DR K=01	SET DR-7 TO CONTINUE BR TESTS.
		BDIA 098	*				IF A STOP OCCURS AT THE ADDRESS
		BDIA 099	*				OF THE NEXT WORD, G1 WAS SET
		BDIA 100	*				WRONG OR THE BRANCH TEST FAILED.
2008	F508	BDIA 101	BCG13N	101	BCG13N	BR IF G1 BIT3=0	G13=1 SHOULD NOT BRANCH
200A	2810	BDIA 102				SET DR K=01	SET DR-7 TO CONTINUE BR TESTS.
200C	A040	BDIA 103		106	UB2040	BR	UNCONDITIONAL BRANCH TO ADR 2040
		BDIA 104	AEND				
		BDIA 105	ATABLE	ADDR=2040			
2040	2810	BDIA 106	UB2040			SET DR K=01	SET DR-7 TO CONTINUE BR TESTS.
		BDIA 107	*				IF A STOP OCCURS AT THE NEXT
		BDIA 108	*				SEQUENTIAL WORD FOLLOWING THIS
		BDIA 109	*				BRANCH, G1 WAS SET WRONG OR THE
		BDIA 110	*				BRANCH TEST FAILED.
2042	C55F	BDIA 111		114	BCG10Y	BR IF G1 BIT0=1	G10=1 SHOULD BRANCH TO ADR 205E
		BDIA 112	AEND				
		BDIA 113	ATABLE	ADDR=205E			
205E	2810	BDIA 114	BCG10Y			SET DR K=01	SET DR-7 TO CONTINUE BR TESTS.
		BDIA 115	*				IF A STOP OCCURS AT THE NEXT
		BDIA 116	*				SEQUENTIAL WORD FOLLOWING THIS
		BDIA 117	*				BRANCH, G1 WAS SET WRONG OR THE
		BDIA 118	*				BRANCH TEST FAILED.
2060	D57F	BDIA 119		125	BCG11Y	BR IF G1 BIT1=1	G11=1 SHOULD BRANCH TO ADR 207E
		BDIA 120	AEND				
		BDIA 121	ATABLE	ADDR=0440			
0440	5007	BDIA 122	DR7OFF 0			STOP	CPU MODE AND DR 7 OFF.
		BDIA 123	AEND				
		BDIA 124	ATABLE	ADDR=207E			
207E	2800	BDIA 125	BCG11Y			SET DR K=00	THIS COMPLETES BDIA BRANCHING
		BDIA 126	*				TESTS.
		BDIA 127	*				*****
		BDIA 128	*				ALU TEST -MANY OF THE WORDS IN THIS TEST CAUSE ALU ERRORS IF THE CKT
		BDIA 129	*				BEING TESTED FAILS AND THEREFORE DO NOT HAVE TO BE TESTED BY BR WDS
		BDIA 130	*				*****
2080	3E09	BDIA 131				H0=0-K00 **FF=00-00 WITH NO CARRY	
2082	C483	BDIA 132	DC0ER	132	DC0ER	BR IF Z=0	CHK FOR DYN BIT 0 BEING OFF.
2084	FC85	BDIA 133	DC7ER	133	DC7ER	BR IF LZ=0	CHK FOR DYN BIT 7 BEING OFF.
2086	E087	BDIA 134	DC6ER	134	DC6ER	BR IF HZ=0	CHK FOR DYN BIT 6 BEING OFF.
2088	F489	BDIA 135	ACONER	135	ACONER	BR IF AC=1	CHK FOR DYN BIT 3 BEING OFF

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
208A	6EE5	BDIA 136				HO=HO\$HO	***FF=FF\$FF
208C	ACDA	BDIA 137		195	CK4FF	BAL	**BRANCH TO CHECK HO FOR FF
208E	2E0D	BDIA 138				HO=HO+K00	***FF=FF+00
		BDIA 139	*				IF A STOP OCCURS AT THE ADDRESS
		BDIA 140	*				OF THE WORD BELOW, THE CHECK FOR
		BDIA 141	*				FF HAS FAILED.
2090	ACDA	BDIA 142		195	CK4FF	BAL	
2092	7EE1	BDIA 143				HO=HO-HO+1	**00=FF-FF+1 WITH A CARRY
		BDIA 144	*				IF A STOP OCCURS AT THE ADDRESS
		BDIA 145	*				OF THE WORD BELOW, THE CHECK FOR
		BDIA 146	*				FF HAS FAILED.
2094	F494	BDIA 147	NOACER	147	NOACER	BR IF AC=0	**SHOULD HAVE DYN 3 BIT ON
2096	7EE1	BDIA 148				HO=HO-HO+1	**00=00-0C+1
2098	6EE3	BDIA 149				HO=HO+HO	**00=00+00
209A	2E5F	BDIA 150				HO=HO+K55	**55=00+55
209C	6EE3	BDIA 151				HO=HO+HO	**AA=55+55
209E	6EE9	BDIA 152				HOC=HO+HO+1	**55=AA+AA+1
20A0	0E5F	BDIA 153				Z=HO\$K55	**00=55\$55
20A2	C4A2	BDIA 154	WT3ER1	154	WT3ER1	BR IF ZNZ	**HO SHOULD BE 55 AND DYN BIT0=1
20A4	2E87	BDIA 155				HO=0\$K88	**88=00\$88
20A6	3E27	BDIA 156				HO=HO\$K22	**AA=88\$22
20A8	2EEF	BDIA 157				HO=HO+KEE	**98=AA+EE
20AA	2E77	BDIA 158				HO=0\$K77	**77=00\$77
20AC	1E87	BDIA 159				HO=HO*-K88	**77=77*-88
20AE	1EB7	BDIA 160				HO=HO*-KBB	**44=77*-BB
20B0	2EF7	BDIA 161				HO=0\$KFF	**FF=00\$FF
20B2	0E11	BDIA 162				Z=HO+K01	**00=FF+01
20B4	F0B4	BDIA 163	DC7ERR	163	DC7ERR	BR IF LZNZ	CHK FOR DYN BIT 7 BEING ON.
20B6	E0B6	BDIA 164	DC6ERR	164	DC6ERR	BR IF HZNZ	CHK FOR DYN BIT 6 BEING ON.
20B8	1EFB	BDIA 165				HO=HO\$K0F	**F0=FF\$0F
20BA	A0E2	BDIA 166		201	CK4F0	BAL	**BR TO CHECK HO FOR F0
20BC	1EF3	BDIA 167				HO=HO*-K0F	**F0=F0*-0F
		BDIA 168	*				IF A STOP OCCURS AT THE ADDRESS
		BDIA 169	*				OF THE WORD BELOW, THE CHECK FOR
		BDIA 170	*				F0 HAS FAILED.
20BE	A0E2	BDIA 171		201	CK4F0	BAL	**BR TO CHECK HO FOR F0
20C0	3E15	BDIA 172				HO=HO\$K10	**F0=F0\$10
		BDIA 173	*				IF A STOP OCCURS AT THE ADDRESS
		BDIA 174	*				OF THE WORD BELOW, THE CHECK FOR
		BDIA 175	*				F0 HAS FAILED.
20C2	A0E2	BDIA 176		201	CK4F0	BAL	**BR TO CHECK HO FOR F0
20C4	3E0D	BDIA 177				HO=HO-K00	**EF=F0-00
		BDIA 178	*				IF A STOP OCCURS AT THE ADDRESS
		BDIA 179	*				OF THE WORD BELOW, THE CHECK FOR
		BDIA 180	*				F0 HAS FAILED.
20C6	2E1D	BDIA 181				HO=HO+K10	**FF=EF+10
20C8	2FF5	BDIA 182				H1=0\$KFO	**F0=00+FO
20CA	A0DA	BDIA 183		195	CK4FF	BAL	**BR TO CHECK HO FOR FF
20CC	0FFD	BDIA 184				Z=H1\$KFO	**00=F0\$FO
		BDIA 185	*				IF A STOP OCCURS AT THE ADDRESS
		BDIA 186	*				OF THE WORD BELOW, THE CHECK FOR
		BDIA 187	*				F0 HAS FAILED.
20CE	C4CE	BDIA 188	ALUER2	188	ALUER2	BR IF ZNZ	**H1 SHOULD BE F0 AND DYN BIT 0=1

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2000	1FF5	BDIA 189				H1=H1*-KFO	**00=F0*-F0
2002	C4D2	BDIA 190	ALUER3	190	ALUER3	BR IF ZNZ	**H1 SHOULD BE ZERO AND DYN BIT 0=1
2004	2EF5	BDIA 191				H0=0\$KFO	**F0=00\$F0
2006	AOE2	BDIA 192		201	CK4F0	BAL	**BR TO CHECK H0 FOR F0
2008	AOE8	BDIA 193		207	MORTST	BR	**BR TO MODE REG SET/RESET TEST
		BDIA 194	*			*****	
20DA	GEFF	BDIA 195	CK4FF			Z=H0\$KFF	H0 SHOULD BE FF
20DC	C4E1	BDIA 196		198	ZISCK	BR IF Z=0	AND DYN BIT 0=1.
20DE	2810	BDIA 197	ANOTF0			SET DR K=01	SET DR 7 AND DO A RTN WHICH
20E0	128E	BDIA 198	ZISOK			RTN	WILL CAUSE A STOP WITH THE
		BDIA 199	*				ADDRESS OF THE BAL TO
		BDIA 200	*				THIS SUBROUTINE + 4.
20E2	0EFD	BDIA 201	CK4F0			Z=H0\$KFO	H0 SHOULD BE F0
20E4	C4DE	BDIA 202		197	ANOTF0	BR IF ZNZ	AND DYN BIT 0=1.
20E6	128E	BDIA 203				RTN	
		BDIA 204	*			*****	
		BDIA 205	*			TEST MODE REG FOR SET/RESET	
		BDIA 206	*			*****	
20E8	24F6	BDIA 207	MORTST			SET MODE K=3F	**SET LS AND EXT MODE TO 7
20EA	578F	BDIA 208				IO=BA	** GO=XIII XIII
20EC	1887	BDIA 209				IO=IO*-K88	** GO=77
20EE	087F	BDIA 210				Z=IO\$K77	
20F0	E0F0	BDIA 211	MDSTHI	211	MDSTHI	BR IF HZNZ	LOOP IF MODE REG 2-4 NOT ALL ON
20F2	F0F2	BDIA 212	MDSTLO	212	MDSTLO	BR IF LZNZ	LOOP IF MODE REG 5-7 NOT ALL ON
20F4	2400	BDIA 213				SET MODE K=00	**SET LS AND EXT MODE TO ZERO
20F6	574F	BDIA 214				GO=BA	**GO=X000X000
20F8	C487	BDIA 215				Z=GO*-K88	**Z=00
20FA	E0FA	BDIA 216	MDRTHI	216	MDRTHI	BR IF HZNZ	LOOP IF MODE REG 2-4 NOT ALL OFF
20FC	F0FC	BDIA 217	MDRTLO	217	MDRTLO	BR IF LZNZ	LOOP IF MODE REG 5-7 NOT ALL OFF
		BDIA 218	*			*****	
		BDIA 219	*			BEGIN STORAGE TEST	
		BDIA 220	*			*****	
20FE	2EF7	BDIA 221				H0=0\$KFF	
2100	2FF7	BDIA 222				H1=0\$KFF	**H=FFFF
2102	6E02	BDIA 223				STH H DC,88	**CTRL ADDR 0388=FFFF
2104	2E05	BDIA 224				H0=0\$K00	
2106	2F05	BDIA 225				H1=0\$K00	**H=0000
2108	4E02	BDIA 226				RDH H DC,88	**READ CTRL STOR 0088 INTO H REG
210A	0FFF	BDIA 227				Z=H1\$KFF	**00=FF\$FF
210C	EC8C	BDIA 228	DCR1ER	228	DCR1ER	BR IF HZNZ	LOOP IF H1 HIGH IS NOT F
210E	F08E	BDIA 229	DCR2ER	229	DCR2ER	BR IF LZNZ	LOOP IF H1 LOW IS NOT F
2110	1EFF	BDIA 230				H0=H0\$KFF	**00=FF\$FF H=00FF
2112	C492	BDIA 231	DCROER	231	DCROER	BR IF ZNZ	**SHOULD HAVE READ FF IN TO H0
2114	2E33	BDIA 232				H0=0\$K03	
2116	2F87	BDIA 233				H1=0\$K88	** H=0388
2118	6FE8	BDIA 234				STB H1 CS,H+1	** CTRL ADDR 0388=88FF H=0389
211A	6FEA	BDIA 235				STB H1 CS,H-1	** CTRL ADDR 0388=8889 H=0388
211C	44E8	BDIA 236				RDH G CS,H+2	** G=8889 H=038A
211E	2F13	BDIA 237				H1=0\$K01	** H=0301
2120	048F	BDIA 238				Z=GO\$K88	**00=88\$88
2122	E0A2	BDIA 239	DCR3ER	239	DCR3ER	BR IF HZNZ	LOOP IF H1 HIGH IS NOT 8
2124	F0A4	BDIA 240	DCR4ER	240	DCR4ER	BR IF LZNZ	LOOP IF H1 LOW IS NOT 8
2126	257F	BDIA 241				G1=G1+K77	**00=89+77

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2128	C4A8	BDIA 242	CSR1ER	242	CSR1ER	BR IF ZNZ	**SHOULD HAVE READ 89 IN TO G1
212A	2EC5	BDIA 243				H0=0\$K00	** H=0001
212C	55EA	BDIA 244				RDB G1 AS,H-1	**H=0000
212E	C4AE	BDIA 245	BMODER	245	BMODER	BR IF ZNZ	**ARITH MODIFIER OFF B REG FAILED IF BR
2130	2E45	BDIA 246				H0=0\$K40	** H=4000
2132	3FE9	BDIA 247				H1=0-KE0	** H=401F
2134	25F7	BDIA 248	AUXSTT			G1=0\$KFF	**G=88FF
2136	75E0	BDIA 249				STB G1 AS,H	**1ST PASS AUX ADDR 401E=XXFF H=401F
		BDIA 250	*				**2ND PASS AUX ADDR 401E=FFFF H=401E
2138	25C5	BDIA 251				G1=0\$K00	**G=8800
213A	55EA	BDIA 252				RDB G1 AS,H-1	**G=88FF AFTER 1ST PASS, H=401E
		BDIA 253	*				** AFTER 2ND PASS, H=401D
213C	05FF	BDIA 254				Z=G10KFF	**00=FF0FF
213E	C48E	BDIA 255	AUXERR	255	AUXERR	BR IF ZNZ	**SHOULD HAVE READ FF IN TO G1
2140	FB34	BDIA 256		248	AUXSTT	BR IF H1 BIT7=0	
2142	44E6	BDIA 257				G=H	** G=401D H=401D DBL BYTE MOD TEST
2144	044D	BDIA 258				Z=G00K40	** 00=40040
2146	C4C6	BDIA 259	DBMERO	259	DBMERO	BR IF ZNZ	** SHOULD HAVE MOVED 40 FROM H0 TO G0.
2148	75F1	BDIA 260				G1=G1-H1+1	** 00=1D-1D+1
214A	C4CA	BDIA 261	DBMER1	261	DBMER1	BR IF ZNZ	** SHOULD HAVE MOVED 1D FROM H1 TO G1
214C	CC60	BDIA 262				RST S K=06	RESET S5 AND S6.
214E	88EC	BDIA 263		269	LSADDR	BR	**BR TO X LINE ADDRESSING TEST
		BDIA 264	AEND				
		BDIA 265	*				*****
		BDIA 266	*				X LINE ADDRESSING
		BDIA 267	*				*****
		BDIA 268	ATABLE		ADDR=08EC		
08EC	D1ED	BDIA 269	LSADDR	269	LSADDR	BR IF S5=1	NO BR. CHECK S5 OFF.
08EE	E1EF	BDIA 270	CKS6OF	270	CKS6OF	BR IF S6=1	NO BR. CHECK S6 OFF.
08F0	2E87	BDIA 271				H0=0\$K88	XXXX ADDRESSABLE 8A AND 8C
08F2	2F05	BDIA 272				H1=0\$K00	X SET TO 8800 WHICH IS
08F4	6E12	BDIA 273				STH H DC,8A	X BR TO 0800
08F6	6E22	BDIA 274				STH H DC,8C	X
08F8	3EE9	BDIA 275				H0=0-KE0	0000 H SET TO 1F77. THIS IS THE
08FA	2F77	BDIA 276				H1=0\$K77	0 INIT VALE REQ'D FOR A=0\$K
08FC	8802	BDIA 277		281	BEGMOD	BR	BR TO 0802
		BDIA 278	AEND				
		BDIA 279	ATABLE		ADDR=0800		
0800	4E02	BDIA 280	MODCWD			RDH H DC,88	XXXX MODIFY CTRL WD AT K88.
0802	2E1B	BDIA 281	BEGMOD			H0=H0+K01	X
0804	2F1D	BDIA 282				H1=H1+K10	X
0806	CA0D	BDIA 283		302	CKPASS	BR IF H0 BIT4=1	X BR AFTER 8TH PASS..
J808	6E02	BDIA 284				STH H DC,88	X
080A	8388	BDIA 285	C			XCTL'8388'	BR TO K ADDR 88 AT ADDR 0388.
		BDIA 286	*				FUNCTION PERFORMED BY CTRL WORDS IN K ADDR. CTRL STORAGE
		BDIA 287	*				FIRST PASS SECOND PASS THIRD PASS FOURTH PASS
		BDIA 288	*			88 2XX7 A=0\$KK	88 0XXF Z=A0KK 88 4XX3 B=AXH+BL 88 0XXF Z=A0KK
		BDIA 289	*				
		BDIA 290	*			2087 U0=0\$K88	008F Z=U0088
		BDIA 291	*			2197 U1=0\$K99	019F Z=U1099
		BDIA 292	*			22A7 V0=0\$KAA	02AF Z=V00AA
		BDIA 293	*			23B7 V1=0\$KBB	03BF Z=V10BB
		BDIA 294	*			24C7 G0=0\$KCC	04CF Z=G00CC

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BDIA 295	*			25D7 G1=0\$KDD 05DF Z=G1\$DD	4553 G1=G1XH+G1L 05DF Z=G1\$DD
		BDIA 296	*			26E7 D0=0\$KEE 06EF Z=D0\$EE	4663 D0=DOXH+DOL 06EF Z=D0\$EE
		BDIA 297	*			27F7 D1=0\$KFF 07FF Z=D1\$FF	4773 D1=D1XH+D1L 07FF Z=D1\$FF
		BDIA 298	*	8A	BR	8A BR IF ZNZ	8A BR 8A BR IF ZNZ
		BDIA 299	*			TO MODCWD. TO 038A.	TO MODCWD TO 038A
		BDIA 300	*	8C	BR	8C BR	8C BR 8C BR
		BDIA 301	*			TO MODCWD. TO MODCWD	TO MODCWD TO MODCWD
080C	E1A3	BDIA 302	CKPASS	313	PASS24	BR IF S6=1	CHECK PASS. BR AFTER 2ND OR 4TH.
080E	2020	BDIA 303				SET S6	S6=1 TO SAY NEXT PASS IS 2 OR 4.
0810	E190	BDIA 304	CKS6E1	304	CKS6E1	BR IF S6=0	CHECK SET OF S6
0812	2EC5	BDIA 305				H0=0\$KCO	XXXX SET 8A TO C48A WHICH IS
0814	3E43	BDIA 306				H0=H0\$K04	X BR IF ZNZ TO 038A
0816	2F85	BDIA 307				H1=0\$K80	X
0818	3FA3	BDIA 308				H1=H1\$K0A	X
081A	6E12	BDIA 309				STH H DC, 8A	X
081C	2EF7	BDIA 310				H0=0\$KFF	0000 H SET TO FF7F. THIS IS THE
081E	3F89	BDIA 311				H1=0-K80	0 INIT VALUE REQ'D FOR Z=AKK
0820	8802	BDIA 312		281	BEGMOD	BR	BR TO BEGIN MODIFIC OF CTRL WD.
0822	0020	BDIA 313	PASS24			RST S6	PASS 2 OR 4 JUST FINISHED. RESET
0824	E1A5	BDIA 314	CKS6E0	314	CKS6E0	BR IF S6=1	S6 TO INDICATE NEXT PASS IS 3RD.
0826	D1AC	BDIA 315		318	NOT4TH	BR IF S5=0	S5=1 INDICATES 4TH PASS COMP.
0828	0040	BDIA 316				RST S5	
082A	83D8	BDIA 317		331	LSSET	BR	BR TO LOCAL STOR SET/RESET TEST
082C	2040	BDIA 318	NOT4TH			SET S5	NEXT PASS IS THIRD
082E	D1AE	BDIA 319	CKS5E1	319	CKS5E1	BR IF S5=0	CK SET OF S5.
0830	4E22	BDIA 320				RDH H DC, 8C	XXXX RESTORE 8A TO BR TO 0800
0832	6E12	BDIA 321				STH H DC, 8A	X
0834	3EC9	BDIA 322				H0=0-KCO	0000 H SET TO 3FF3. THIS IS INIT
0836	5EF1	BDIA 323				H1=H0X	0 VALUE REQ'D FOR B=AXH+BL
0838	8802	BDIA 324		281	BEGMOD	BR	BR TO BEGIN MODIFIC OF CTRL WD.
		BDIA 325	AEND				
		BDIA 326	*				
		BDIA 327	*				
		BDIA 328	*			*****	
		BDIA 329	*			LOCAL STORAGE SET/RESET ROUTINE..... ZONE ZERO	
		BDIA 329	*			*****	
		BDIA 330	ATABLE		ADDR=03D8		
03D8	4812	BDIA 331	LSSET			RDH I DC, 8A	I=C48A XXX K-ADDR BC IS SET TO
03DA	3935	BDIA 332				I1=I1\$K30	I=C48A X C48C WHICH IS A
03DC	2928	BDIA 333				I1=I1\$K02	I=C48C X BR IF ZNZ
03DE	68E2	BDIA 334				STH I DC, BC	X TO ITSELF.
03E0	2A25	BDIA 335				T0=0\$K20	XXXX T SET TO 20F7 WHICH IS
03E2	2BF5	BDIA 336				T1=0\$KFO	X UO=0\$KFF
03E4	3B73	BDIA 337				T1=T1\$K07	X
03E6	2C25	BDIA 338				P0=0\$K20	0000 P SET TO 201B WHICH IS
03E8	2D15	BDIA 339				P1=0\$K10	0 UO=UO\$K01
03EA	3DB3	BDIA 340				P1=P1\$K0B	0
03EC	2EC5	BDIA 341				H0=0\$K00	XXXX H SET TO 0001 WHICH IS
03EE	2F13	BDIA 342				H1=0\$K01	X Z=UO\$K00
03FO	83D0	BDIA 343		355	STNWCW	BR	BR TO STORE THE THREE CONTROL
		BDIA 344	*				WORDS JUST BUILT IN T, P, AND H.
		BDIA 345	AEND				
		BDIA 346	ATABLE		ADDR=03C0		
03C0	C4C0	BDIA 347	NOTRST	347	NOTRST	BR IF ZNZ	CK THAT ALL BITS WERE RESET.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
03C2	4AC2	BDIA 348	RDOLCW			RDH T DC, B8	READ OLD CTRL WORDS.
03C4	4CD2	BDIA 349				RDH P DC, BA	READ OLD CTRL WORDS.
03C6	4EF2	BDIA 350				RDH H DC, BE	READ OLD CTRL WORDS.
03C8	2A1B	BDIA 351				TO=TO+K01	MODIFY CTRL WDS...INCREMENT X.
03CA	2C1B	BDIA 352				PO=PO+K01	MODIFY CTRL WDS...INCREMENT X.
03CC	2E1B	BDIA 353				HO=HO+K01	MODIFY CTRL WDS...INCREMENT X.
03CE	E0F2	BDIA 354		373	DYNCR	BR IF HZNZ	BR IF SIXTEENTH PASS COMPLETED.
03D0	6AC2	BDIA 355	STNWCW			STH T DC, B8	STORE NEW CONTROL WORDS.
03D2	6CD2	BDIA 356				STH P DC, BA	STORE NEW CONTROL WORDS.
03D4	6EF2	BDIA 357				STH H DC, BE	STORE NEW CONTROL WORDS.
03D6	83B8	BDIA 358	C			XCTL'83B8'	BR TO K-ADDRESSABLE B8.
		BDIA 359	AEND				
		BDIA 360	*				*****
		BDIA 361	*				* THE 4 CCNTROL WORDS BELOW WERE BUILT IN K-ADDRESSABLE CONTROL
		BDIA 362	*				* STORAGE AND BRANCHED TO FROM THE WORD IMMEDIATLY ABOVE THESE COMENTS.
		BDIA 363	*				* THIS IS DONE 16 DIFFERENT TIMES, WITH 'X' INCREMENTED EACH TIME TO
		BDIA 364	*				* MODIFY THE FUNCTION OF THE WORDS.
		BDIA 365	*				* K-ADDR B8...CTL STOR ADDR 03B8 EQUALS 2XF7 A=A\$KFF
		BDIA 366	*				* K-ADDR BA...CTL STOR ADDR 03BA EQUALS 2X1B A=A+K01
		BDIA 367	*				* K-ADDR BC...CTL STOR ADDR 03BC EQUALS C4BC BR IF ZNZ TO ITSELF
		BDIA 368	*				* K-ADDR BE...CTL STOR ADDR 03BE EQUALS 0X01 Z=A+K00
		BDIA 369	*				*
		BDIA 370	*				*****
		BDIA 371	ATABLE ADDR=03F2				
		BDIA 372	*				*****
03F2	2505	BDIA 373	DYNCR			G1=0\$K00	**SET G1 TO ZERO AND THEN INCREMENT TO
03F4	2518	BDIA 374	ZINCR			G1=G1+K01	** OBTAIN ALL BIT COMBINATIONS ON Z BUS
03F6	F4FD	BDIA 375		381	ALUEDT	BR IF AC=1	**END OF TEST--BR TO ALU OE TEST
03F8	C4F4	BDIA 376		374	ZINCR	BR IF ZNZ	**BR BACK TO CHECK ALL Z BUSS BIT COMB
03FA	5007	BDIA 377				STOP	**DYN COND REG BIT 0 IS ON IN ERROR
		BDIA 378	*				*****
		BDIA 379	*				ALU ERROR DETECTION TEST
		BDIA 380	*				*****
03FC	2413	BDIA 381	ALUEDT			GO=0\$K01	
03FE	883C	BDIA 382		386	ZEROG1	BR	
		BDIA 383	AEND				
		BDIA 384	ATABLE ADDR=083A				
082A	6443	BDIA 385	CKALOE			GO=GO+GO	
083C	25C5	BDIA 386	ZEROG1			G1=0\$K00	
083E	7543	BDIA 387				G1=G1-G0	
0840	38C8	BDIA 388				SET DR K=C0	**DISABLE STOP ON ERROR & FORCE PLUS
0842	050D	BDIA 389				Z=G1\$K00	** SIDE OF ALU A ENTRY CAUSING ALU CK
0844	3800	BDIA 390				SET DR K=80	
0846	DAC6	BDIA 391	ALCKER	391	ALCKER	BR IF MC BIT5=0	**ALU ERROR SHOULD HAVE SET MC 5
0848	1212	BDIA 392				RST MMSK K=91	
084A	2800	BDIA 393				SET DR K=00	
084C	C43A	BDIA 394		385	CKALOE	BR IF GO BIT0=0	**BRANCH BACK TO CHECK EACH OE CKT
		BDIA 395	*				*****
		BDIA 396	*				STORAGE DATA, CONTROL WORD, STORAGE ADDRESS, A REG AND B REG
		BDIA 397	*				PARITY DETECTION TEST----DC REG BITS 3,6 AND 7 MUST BE OFF
		BDIA 398	*				*****
084E	2440	BDIA 399				SET MODE K=04	
0850	886E	BDIA 400		415	CKSTPC	BAL	**STORE ADDR OF CHECK DATA TO BE READ

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0852	0E3F	BDIA 401				Z=H0M K33	**THIS ADDR CONTAINS 0E3F
0854	FED4	BDIA 402	ROCRCE	402	ROCRCE	BR IF MC3=0	**SHOULD HAVE RO CTRL WORD PARITY CHECK
0856	4E80	BDIA 403				RDH H CS,I	**H=0E3F WITH EVEN PARITY IN H0
0858	6E80	BDIA 404				STH H CS,I	**RESTORE CHECK DATA WITH GOOD PARITY
085A	FACA	BDIA 405	NOBPC	405	NOBPC	BR IF MC7=0	**SHOULD HAVE SET B REG PARITY CK LATCH
085C	0E3F	BDIA 406				Z=H0M K33	**SET A REG PARITY CK LT & RST DC3,6&7
085E	EAD E	BDIA 407	NOAPC	407	NOAPC	BR IF MC6=0	**SHOULD HAVE SET A REG PARITY CK LATCH
0860	1212	BDIA 408				RST MMSK K=91	**RESET MC REG
0862	886E	BDIA 409		415	CKSTPC	BAL	**STORE ADDR OF CHECK DATA TO BE READ
0864	0F1F	BDIA 410				Z=H1M K11	**THIS ADDR CONTAINS 0F1F
0866	FE E6	BDIA 411	R1CRCE	411	R1CRCE	BR IF MC3=0	**SHOULD HAVE R1 CTRL WORD PARITY CHECK
0868	4E80	BDIA 412				RDH H CS,I	**H=0F1F WITH EVEN PARITY IN H1
086A	6E80	BDIA 413				STH H CS,I	**RESTORE CHECK DATA WITH GOOD PARITY
086C	02E2	BDIA 414				RTN H MMSK1=0	**CAUSE STOR ADDR CK--H1=1F WITH P BIT
086E	4E80	BDIA 415	CKSTPC			RDH H CS,I	**1ST LOOP H=0E3F,ON 2ND LOOP H=0F1F
0870	3804	BDIA 416				SET DR K=A0	**DISABLE STOP ON ERROR AND FORCE
0872	6E80	BDIA 417				STH H CS,I	** STORE BITS P0 AND P1 TO MEMORY
0874	CAF4	BDIA 418	NOSDPC	418	NOSDPC	BR IF MC4=0	**SHOULD HAVE SET STOR DATA PARITY LT
0876	3800	BDIA 419				SET DR K=80	**RST FORCING OF STORE BITS P0 AND P1
0878	128E	BDIA 420				RTN	**RTN FOR ADDITIONAL TESTING & 2ND LOOP
		BDIA 421	AEND				
		BDIA 422	ATABLE	ADDR=0F1E			
0F1E	EE9E	BDIA 423	NOSTAC	423	NOSTAC	BR IF MC2=0	**SHOULD HAVE SET STORAGE ADDRESS CHECK
0F20	1212	BDIA 424				RST MMSK K=91	**RESET MC REG
0F22	5EFF	BDIA 425				H1=MC	**CHECK RESET OF MC REG
0F24	04A4	BDIA 426	MCRSTE	426	MCRSTE	BR IF ZNZ	**MC REG SHOULD HAVE BEEN RESET
0F26	2800	BDIA 427				SET DR K=00	**ALLOW STOP ON ERROR
0F28	2020	BDIA 428				SET S6	
0F2A	8024	BDIA 432		BCHK 005	START	BR	
		BDIA 438	AEND				

 * CROSS REFERENCE FOR CSECT BDIA *

- BDIA 018
- BDIA 056
- BDIA 061
- BDIA 066
- BDIA 075
- BDIA 083
- BDIA 087
- BDIA 096
- BDIA 101
- BDIA 106
- BDIA 114
- BDIA 122
- BDIA 125
- BDIA 132
- BDIA 133
- BDIA 134
- BDIA 135
- BDIA 147
- BDIA 154
- BDIA 163

- BDIA 008
- BDIA 053
- BDIA 061
- BDIA 066
- BDIA 072
- BDIA 080
- BDIA 084
- BDIA 096
- BDIA 101
- BDIA 103
- BDIA 111
- BDIA 054
- BDIA 119
- BDIA 132
- BDIA 133
- BDIA 134
- BDIA 135
- BDIA 147
- BDIA 154
- BDIA 163

 * CROSS REFERENCE FOR CSECT BDIA *

BDIA 164	BDIA 164		
BDIA 188	BDIA 188		
BDIA 190	BDIA 150		
BDIA 195	BDIA 137	BDIA 142	BDIA 183
BDIA 197	BDIA 202		
BDIA 198	BDIA 156		
BDIA 201	BDIA 166	BDIA 171	BDIA 176 BDIA 192
BDIA 207	BDIA 193		
BDIA 211	BDIA 211		
BDIA 212	BDIA 212		
BDIA 216	BDIA 216		
BDIA 217	BDIA 217		
BDIA 228	BDIA 228		
BDIA 229	BDIA 229		
BDIA 231	BDIA 231		
BDIA 239	BDIA 239		
BDIA 240	BDIA 240		
BDIA 242	BDIA 242		
BDIA 245	BDIA 245		
BDIA 248	BDIA 256		
BDIA 255	BDIA 255		
BDIA 259	BDIA 259		
BDIA 261	BDIA 261		
BDIA 269	BDIA 263	BDIA 269	
BDIA 270	BDIA 270		
BDIA 281	BDIA 277	BDIA 312	BDIA 324
BDIA 302	BDIA 283		
BDIA 304	BDIA 304		
BDIA 313	BDIA 302		
BDIA 314	BDIA 314		
BDIA 318	BDIA 315		
BDIA 319	BDIA 319		
BDIA 331	BDIA 317		
BDIA 347	BDIA 347		
BDIA 355	BDIA 343		
BDIA 373	BDIA 354		
BDIA 374	BDIA 376		
BDIA 381	BDIA 375		
BDIA 385	BDIA 394		
BDIA 386	BDIA 382		
BDIA 391	BDIA 391		
BDIA 402	BDIA 4C2		
BDIA 405	BDIA 4C5		
BDIA 407	BDIA 4C7		
BDIA 411	BDIA 411		
BDIA 415	BDIA 4C0	BDIA 409	
BDIA 418	BDIA 418		
BDIA 423	BDIA 423		
BDIA 426	BDIA 426		

BMCK DESCRIPTIVE TEXT

WHEN A MACHINE ERROR OCCURS (SEE -MC- REGISTER), A MACHINE CHECK TRAP IS TAKEN IF -
MMSK 8 AND 9 NOT ON
MACHINE CHECK MASK LATCH ON (MW BIT 5)

IF MMSK 8 OR 9 ARE ON AND A MACHINE ERROR OCCURS, THE HARD STOP LATCH IS SET AND THE SYSTEM STOPS.

IF THE MACHINE CHECK TRAP IS ALLOWED THE TRAP ADDRESS 0220 IS FORCED ON THE SAR LINES AND THE -BMCK- ROUTINE IS ENTERED AT LABEL * TRAPWD *.

MMSK BITS 0-6 ARE TESTED AND IF ANY ARE ON (INDICATES THAT AN I/O TRAP OPERATION WAS IN PROGRESS WHEN THE MACHINE CHECK OCCURED), THE I/O OPERATION WILL BE TERMINATED, AND AN ATTEMPT TO CAUSE AN I/O INTERRUPT IS MADE. IF MMSK BITS 0-6 ARE NOT ON, A MACHINE CHECK INTERRUPT TAKES PLACE.

SINCE THE LOGOUT LATCH IS SET IN THIS ROUTINE, NO CPU INSTRUCTION WILL BE EXECUTED UNTIL THE PRINTOUT OF THE LOGOUT AREA IS COMPLETED. ANY PREVIOUS PR-KB OPERATION IN PROCESS, SUCH AS ALTER/DISPLAY, INSTRUCTION STEP ADDRESS TYPE- OUT, OR NORMAL PR-KB FUNCTIONS WILL BE FORCED TO TERMINATE. THE PRINT OUT OF THE LOGOUT AREA STARTS AFTER A CARRIAGE RETURN WITHOUT ANY HEADER INFORMATION.

THE -BCHK- ROUTINE LOADS THE LOGOUT AREA IN PROGRAM STORAGE, LOCATIONS 0080 - 0085, WITH THE INFORMATION LISTED BELOW. THE -ALDP- ROUTINE HANDLES THE PRINTOUT OF THIS LOGOUT AREA ON THE PR-KB.

MACHINE CHECK LOGOUT AREA

LOC 0080	LOC 0081	LOC 0082	LOC 0083	LOC 0084-0085
TRAP PRIORITY REGISTER (MMSK)	BRANCH CONDITION REGISTER (BA)	MACHINE CHECK REGISTER (MC)	ERROR COUNT	BACKUP ADDRESS-
BIT	BIT	BIT		ADDRESS OF THE CONTROL WORD IN OPERATION WHEN THE MACHINE CHECK OCCURRED.
0- CHANNEL HIGH PRIORITY TRAP.	0- CHANNEL 0 INTERRUPT LATCH	0- FILE CONTROL CHECK		
1- 2311 DISK CONTROL TRAP	1- MODE BIT 0	1- STORAGE PROTECT PARITY CHECK		
2- CHANNEL LOW PRIORITY TRAP	2- MODE BIT 1	2- STORAGE ADDRESS PARITY CHECK		
3- 2540 READER TRAP	3- MODE BIT 2	3- CONTROL WORD PARITY CHECK		
4- 2540 PUNCH TRAP	4- IPL LATCH	4- STORAGE DATA PARITY CHECK		
5- COMM. BIT SERVICE TRAP	5- LS ZONE BIT 0	5- ALU ERROR CHECK		
6- COMM. CHAR. SERVICE TRAP	6- LS ZONE BIT 1	6- A-REG PARITY CHECK		
7- LEVEL 1 PRIORITY HOLD	7- LS ZONE BIT 2	7- B-REG PARITY CHECK		

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BMCK 001	T			MACHINE CHECK TRAP ROUTINE	
		BMCK 002	ATABLE			ADDR=0220	
0220	3262	BMCK 003	TRAPWD			LINK D MMSK9=1	SAVE ADDR+2 OF FAILING MICRO WD
0222	8CA6	BMCK 004		006	GETOUT	BR	GET OUT OF TRAP AREA
		BMCK 005	AEND				
OCA6	7612	BMCK 006	GETOUT			STH D DA,8A	TEMP STORE D REG FOR CHNL
OCA8	566F	BMCK 007				DO=MMSK	MOVE MMSK (TRAP PRIORITY) REG
OCAA	577F	BMCK 008				D1=BA	MOVE BA(BRANCH CONDITION) REG
OCAC	7672	BMCK 009				STH D DA,9E	SAVE MMSK & BA REGS IN AUX K-7
OCAE	2610	BMCK 010				SET BC K=01	SET LOGOUT LATCH
OCB0	A2CF	BMCK 011		028	INTRAP N	BR IF MMSK=NZ	BR IF ANY MMSK BIT

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
JCB2	2400	BMCK 012	NOTRAP			SET MODE K=00	SET TO CPU MODE & ZONE
JCB4	2135	BMCK 018				U1=0\$K30	MACHINE CHECK OLD PSW ADDR
JCB6	98FA	BMCK 019		CSTS 095	PICKUP	BR	GO TO STORE & LOAD MC PSW
JF40	C6D3	BMCK 028	INTRAP 0	037	CHNLCK	BR IF MMSK0=1	BR IF IN CHNL HI PRIORITY TRAP
JF42	D6D5	BMCK 029		038	DSKCHN	BR IF MMSK1=1	BR IF IN DISK CHAINING TRAP
JF44	E6D3	BMCK 030		037	CHNLCK	BR IF MMSK2=1	BR IF IN CHNL LO PRIORITY TRAP
JF46	F6D7	BMCK 031		039	RDRTRP	BR IF MMSK3=1	BR IF IN READER TRAP
JF48	C2DB	BMCK 032		041	PCHTRP	BR IF MMSK4=1	BR IF IN PUNCH TRAP
JF4A	D2DF	BMCK 033		043	COMBIT	BR IF MMSK5=1	BR IF IN COMM. BIT SVC TRAP
JF4C	E2E1	BMCK 034		044	COMBYT	BR IF MMSK6=1	BR IF IN COMM. CHAR SVC TRAP
JF4E	021E	BMCK 035	RSTK7			RST MMSK K=71	RST LEVEL 1 PRIORITY
JF50	8CB2	BMCK 036		012	NOTRAP	BR	LEVEL 1 PRIORITY, NOT IN TRAP
JF52	83BC	BMCK 037	CHNLCK	DCLR 196	CCCXXX	BR	CHANNEL TRAPS
JF54	94AC	BMCK 038	DSKCHN	FILT 075	MCHCHK	BR	
JF56	2D02	BMCK 039	RDRTRP			SET RP K=10	SET READER MACHINE CK LATCH
JF58	8EF6	BMCK 040		ETTR 043	RQUEST	BR	GO BACK TO RDR TRAP ROUTINE
JF5A	2D40	BMCK 041	PCHTRP			SET RP K=04	SET PUNCH MACHINE CK LATCH
JF5C	8FF2	BMCK 042		ETRP 066	INLK	BR	GO BACK TO PCH TRAP ROUTINE
JF5E	021A	BMCK 043	COMBIT			RST MMSK K=51	RST BIT SERVICE TRAP
JF60	021C	BMCK 044	COMBYT			RST MMSK K=61	RST GENERAL ICA TRAP
JF62	1D00	BMCK 045				RST CSETF K=80	ICA RESET
JF64	8F4E	BMCK 046		035	RSTK7	BR	
137A	2C07	BMCK 047	LOGOUT			U0=0	
137C	2400	BMCK 048				SET MODE K=00	SET CPU MODE TO BR ON MC REG
137E	5E3F	BMCK 049				V1=MC	MOVE MC REG
1380	C4A7	BMCK 050		075	NOTMC	BR IF Z=0	BR IF NOT MACHINE CK LOGOUT
1382	1212	BMCK 051				RST MMSK K=91	RST MACHINE CHECK PRIORITY
		BMCK 052	*				
		BMCK 053	*				
		BMCK 054	*				
		BMCK 055	*				
		BMCK 056	*				
		BMCK 057	*				
						FROM THIS POINT ON, A SECOND MACHINE CHECK WOULD CAUSE A RETURN TO THE TRAP ADDRESS 0220. IF THE SECOND MACHINE CHECK OCCURRED PRIOR TO THE LAST INSTRUCTION, THE CPU WILL STOP WITH THE CLOCK OFF. THE LAST INSTRUCTION ALSO RESETS THE MC REGISTER.	
1384	5672	BMCK 058				RDH D DA,9E	RD SAVED MMSK & BA FROM AUX K-7
1386	2440	BMCK 059				SET MODE K=04	BACK UP ZONE
1388	7672	BMCK 060				STH D DA,9E	XFER MC BACKUP ADDR TO AUX K-7
138A	24C0	BMCK 061				SET MODE K=00	SET TO CPU MODE & ZONE
138C	2185	BMCK 062				U1=0\$K80	SET UP ADDR OF LOGOUT AREA
138E	420F	BMCK 063				STPO=U0	DISABLE STORAGE PROTECT
1390	7618	BMCK 064				STH D U+2	STORE MMSK, BA IN LOCATION 80,81
1392	7318	BMCK 065				STB V1 U+1	STORE MACHINE CK REG IN 82
1394	571C	BMCK 066				RDB D1 U+0	READ LAST ERROR COUNT FROM 83
1396	271B	BMCK 067				D1=D1+K01	ADD 1 TO ERROR COUNT
1398	7718	BMCK 068				STB D1 U+1	STORE UPDATED ERROR COUNT IN 83
139A	5672	BMCK 069				RDH D DA,9E	READ BACKUP MICRO ADDR INTO D
139C	16C5	BMCK 070				DO=DO*-KCO	STRIP BITS 0,1 - HZ,LZ
139E	1713	BMCK 071				D1=D1*-K01	STRIP BIT 15 - ALU CARRY
13A0	6666	BMCK 072				D=D-2	MICRO ADDR POINTS AT FAILING WD
13A2	7618	BMCK 073				STH D U+2	MICRO WD ADDR OF FAILURE IN 84,5
13A4	A7FA	BMCK 074		CCOM 181	RSTRKY	BAL	RESTORE THE CPU KEY
13A6	2404	BMCK 075	NOTMC			SET MODE K=20	SET BACK TO 1052 MODE
13A8	2233	BMCK 076				VO=0\$K03	HW COUNT EQUALS 3
13AA	1F2E	BMCK 077				RST TA K=F2	**RST RD, WT, M-FORCE, ALDP, SHARE IF ON

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
13AC	2171	BMCK 078				U1=0-K07	U=COF8 - PRKB(1052) UCW ADDR
13AE	2807	BMCK 079				T1=0	
13B0	7A00	BMCK 080				STH T AS,U+0	RST 1ST HW OF 1052 UCW
13B2	4632	BMCK 081				RDH D DC,8E	READ OUT CONSTANT OF 4E08
13B4	261D	BMCK 082				DO=DO+K10	SET D TO 5E18 FOR
13B6	271D	BMCK 083				D1=D1+K10	'RDH H U+2'
13B8	6662	BMCK 084				STH D DC,9C	STORE 'DOIT' WD FOR ALDP ROUTINE
13BA	17AB	BMCK 085				D1=D1+K0A	SET FLAGS FOR MC CK OR CE TRAP
13BC	2185	BMCK 086				U1=0\$K80	SET UP ADDR
13BE	873C	BMCK 087		ALDP 182	LINEFD	BAL	FOR A LINE FEED
13C0	968C	BMCK 088		ALDP 100	COLCNT	BR	GO TO PRINT OUT LOG AREA

 * CROSS REFERENCE FOR CSECT BMCK *

BMCK 003	FILT 074	
BMCK 006	BMCK 004	
BMCK 012	BMCK 036	
BMCK 028	BMCK 011	
BMCK 035	BMCK 046	
BMCK 037	BMCK 028	BMCK 030
BMCK 038	BMCK 029	
BMCK 039	BMCK 031	
BMCK 041	BMCK 032	
BMCK 043	BMCK 033	
BMCK 044	BMCK 034	
BMCK 047	DYPE 021	
BMCK 075	BMCK 050	

BPSW DESCRIPTIVE TEXT

ENTRY POINTS TO THE BPSW ROUTINE

PROG WHEN A PROGRAM INTERRUPTION IS DETECTED DURING INSTRUCTION PROCESSING THE BPSW ROUTINE IS ENTERED HERE. THE CURRENT PSW, ALONG WITH THE INTERRUPTION CODE, IS STORED IN THE PROGRAM OLD PSW LOCATION AND THE PROGRAM NEW PSW IS READ OUT AND PLACED IN CONTROL.

PROGA ENTRY IS MADE HERE IF A PROTECTION ERROR OR AN EXECUTE INSTRUCTION IS DETECTED. PSW HANDLING IS THE SAME AS FOR PROG ENTRY.

ENTRYB ENTRY IS MADE HERE FOR EXTERNAL INTERRUPT, CHANNEL INTERRUPT, OR CHANNEL INITIAL SELECTION. HIGH ORDER HEX DIGIT FOR PROPER PSW ADDRESS IS SET PRIOR TO ENTRY HERE.

ENTRY A SUPERVISOR CALL OR MACHINE CHECK TRAP CAUSES ENTRY AT THIS LABEL.

LPSW ENTRY HERE IS TO COMPLETE THE 360 LOAD PSW INSTRUCTION OR TO LOAD THE IPL PSW.

INTBR ENTRY HERE IS FROM CONDITIONAL BRANCH INSTRUCTIONS TO CHECK FOR ADDRESSING OR SPECIFICATION ERRORS.

INTXA
NINTXA ENTRY TO EITHER OF THESE LABELS IS FROM THE BSWI ROUTINE WHEN THE SOFT STOP LOOP IS EXITED.

WAIT THIS LABEL IS BRANCHED TO WHEN THE WAIT BIT IS TESTED AND FOUND TO BE ON.

PROGRAM STATUS WORD AND THE LOCATION OF THE CURRENT PSW

0	7 8	11 12	15 16	31 32	33 34	35 36	39 40	47 48	63

* SYSTEM MASK	* KEY	* AMWP	* INTERRUPTION CODE	* ILC	* CC	* MASK	* PROG	* INSTRUCTION ADDRESS	*

. AUX	. AUX	. AUX	.	. LS	. LS	. LS	.	. AA	. AUX
. A8	. A9	. A9	.	. GO	. PO	. P1	.	. AB	.
.0	7.0	3.4	7.	. .0	3.0	3.	.	.	.
.....									

PSW ADDRESS ASSIGNMENTS

HEX		HEX	
00	INITIAL PROGRAM LOADING PSW		
18	EXTERNAL OLD PSW	58	EXTERNAL NEW PSW
20	SUPERVISOR CALL OLD PSW	60	SUPERVISOR CALL NEW PSW
28	PROGRAM OLD PSW	68	PROGRAM NEW PSW
30	MACHINE-CHECK OLD PSW	70	MACHINE-CHECK NEW PSW
38	INPUT/OUTPUT OLD PSW	78	INPUT/OUTPUT NEW PSW

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BPSW 001	T			PSW LOAD AND STORING	001
		BPSW 002	*				
1C70	2F07	BPSW 003	PROG			H1=0	CLEAR INVALID ADD.
1C72	2125	BPSW 004	PROGA			U1=0\$K20	CREATE
1C74	21E8	BPSW 005	ENTRYB			U1=U1+K0E	PROG. PSW
1C76	20G7	BPSW 006	ENTRY			U0=0	ADDRESS
1C78	420F	BPSW 007	ENTRYA			STPO=U0	STOR PROT DISABLE
1C7A	5CEB	BPSW 008				H0=POH	PLACE THE 4 BIT CC INTO H0 HIGH
1C7C	3E45	BPSW 009				H0=H0\$K40	ASSURE ILC = 1
1C7E	4DE5	BPSW 010				H0=PIXL\$H0H	PLACE PROG MASK INTO H0 LOW
1C80	E187	BPSW 011		014	NOTEXC	BR IF S6=1	TEST FOR EXECUTE
1C82	5892	BPSW 012				RDH I DA, AA	READ OUT EXECUTE INST. COUNTER
1C84	5C8C	BPSW 013		017	EXECA	BR	
1C86	C40A	BPSW 014	NOTEXC	016	WOW	BR IF G00=0	TEST OPCODE FORMAT FOR LENGTH
1C88	2E4D	BPSW 015				H0=H0+K40	ADD 1 TO LENGTH CODE ILC = 2
1C8A	D40E	BPSW 016	WOW	018	EXEC	BR IF G01=0	TEST OPCODE FORMAT FOR LENGTH
1C8C	2E4D	BPSW 017	EXECA			H0=H0+K40	ADD 1 TO LENGTH CODE ILC=2 OR 3
1C8E	781A	BPSW 018	EXEC			STH I U-2	STORE INSTRUCTION ADDRESS
1C90	7E1A	BPSW 019				STH H U-2	STORE ILC, CC, AND PROG. MASK
1C92	7A1A	BPSW 020				STH T U-2	STORE INTERRUPT CODE
1C94	5E82	BPSW 021				RDH H DA, A8	READOUT SYS MASK, KEY, AMWP
1C96	7E10	BPSW 022				STH H U	STORE INTO OLD PSW
1C98	214D	BPSW 023				U1=U1+K40	ADJUST TO NEW PSW ADDRESS
1C9A	5C18	BPSW 024	LPSW			RDH P U+2	START NEW PSW READ
1C9C	44CF	BPSW 025				SM=PO	SET SYSTEM MASK
1C9E	42DF	BPSW 027				STPO=P1	SET PROTECT KEY
1CA0	7C82	BPSW 029				STH P DA, A8	STORE SYS MASK, KEY, AND AMWP
1CA2	6004	BPSW 030				U=U+2	POINT TO 3RD HALFWORD OF PSW
1CA4	5E18	BPSW 031				RDH H U+2	PLACE 3RD HALFWORD PSW IN H-REG
1CA6	5810	BPSW 032				RDH I U	PLACE 4TH HALFWORD PSW IN I-REG
1CA8	4ED3	BPSW 033				P1=HOXH+P1L	PLACE PROG MASK AND AMWP INTO P1
1CAA	5ECB	BPSW 034				PO=H0H	PLACE ILC AND CC INTO PO HIGH
1CAC	1CC5	BPSW 035				PO=PO*-KCO	ZERO OUT ILC
1CAE	FC32	BPSW 036		038	NOT	BR IF P03=0	BRANCH IF CC IS 0 OR 2
1CB0	3C45	BPSW 037				PO=PO\$K40	MAKE 4 BIT CC = TO 5 OR 7
1CB2	2A07	BPSW 038	NOT			T0=0	
1CB4	7CC2	BPSW 039				STH P DA, B8	SAVE PO-1
1CB6	4FDF	BPSW 040				MW=P1	SET WAIT & MACHINE CK BITS
1CB8	E947	BPSW 041	INTXB	062	WAIT	BR IF P16=1	BRANCH IF WAIT BIT ON
1CBA	2020	BPSW 053	INTBR			SET S6	SET EXECUTE INDICATOR OFF
1CBC	F1CA	BPSW 054		064	INTX	BR IF S7=0	BRANCH IF EXCEPTIONAL CONDITION
1CBE	AB07	BPSW 055	INTXA	BPSW 065	ADDRER N	BR IF H1=NZ	BRANCH IF 24 BIT ADDRESS
1CC0	09E3	BPSW 056				Z=I1*-K0E	TEST LOW ADDRESS BYTE
1CC2	F0CC	BPSW 057		059	SPEC	BR IF LZNZ	BRANCH IF ODD ADDRESS
1CC4	A9A6	BPSW 058		CICY 008	ISTART 3	BR	BRANCH TO ICYCLES FOR INST. READ
1CCC	2407	BPSW 059	SPEC			GO=0	ZERO OP CODE REGISTER
1CCE	2B63	BPSW 060				T1=0\$K06	SET BITS TO IDENTIFY SPEC. CHECK
1CDO	9B04	BPSW 061		067	PROG1	BR	BRANCH TO INCREMENT INST. ADDR.
1CC6	0004	BPSW 062	WAIT			RST S2	** WAIT LOOP
1CC8	F1C7	BPSW 063		062	WAIT	BR IF S7=1	** LEAVE WAIT LOOP ON INTERRUPT
1CCA	87CA	BPSW 064	INTX	BSWI 008	CHECKE	BR	BRANCH TO DETERMINE INTERRUPT
1B00	2405	BPSW 065	ADDRER 0			GO=0\$K00	ZERO OP CODE REGISTER
1B02	2B53	BPSW 066				T1=0\$K05	SET BITS TO IDENTIFY ADDR. ERROR

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1B04	6884	BPSW 067	PROG1			I=I+2	INCREMENT INSTRUCTION ADDRESS
1B06	9C72	BPSW 068		BPSW 004	PROGA	BR	BRANCH TO STORE INTERRUPT CODE

* CROSS REFERENCE FOR CSECT BPSW *							

BPSW 003	CCOM 072	COMD 061	CFAD 132	CFHA 037	CICY 046	CICY 060	CICY 067 CICY 069 CLST 015 CNVR 121 CSAS 130
BPSW 004	BPSW 068	BWRP 069	CBRC 063				
BPSW 005	BSWI 090	DCHN 076					
BPSW 006	CSTS 057						
BPSW 014	BPSW 011						
BPSW 016	BPSW 014						
BPSW 017	BPSW 013						
BPSW 018	BPSW 016						
BPSW 024	BSYS 128	CSTS 007					
BPSW 038	BPSW 036	BSYS 137					
BPSW 053	CBRC 016						
BPSW 055	BSWI 135						
BPSW 059	BPSW 057						
BPSW 062	BPSW 041	BPSW 063	BSWI 049	BSWI 139			
BPSW 064	BPSW 054						
BPSW 065	BPSW 055						
BPSW 067	BPSW 061						

BSTP DESCRIPTIVE TEXT

ENTRY TO THE -BSTP- ROUTINE IS MADE FROM THE -BSWI- ROUTINE WHEN AN INSTRUCTION STEP OPERATION OR A SET IC OPERATION IS PERFORMED. THE ENTRY IS MADE TO TEST THE 1052 AND DETERMINE IF THE INSTRUCTION COUNTER CAN BE TYPED OUT.

IF THE 1052 IS NOT BUSY, THE -ALDP- ROUTINE IS BRANCHED TO, AND THE INSTRUCTION COUNTER IS TYPED OUT.

IF THE 1052 IS BUSY, THE INSTRUCTION COUNTER IS NOT TYPED OUT. THE -BSWI- ROUTINE IS BRANCHED TO. THE CPU KEY, P-REG, AND I-REG ARE RESTORED AND THE -CICY- ROUTINE IS BRANCHED TO.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BSTP 001	T			INSTRUCTION TYPEOUT	
0F06	2404	BSTP 002	TYPADD			SET MODE K=20	
0F08	CF8F	BSTP 003		006	BUSY	BR IF TU0=1	CHECK AND BRANCH IF READ LATCH
0F0A	DF8F	BSTP 004		006	BUSY	BR IF TU1=1	CHECK AND BRANCH IF WRT LATCH
0F0C	CE90	BSTP 005		007	OK	BR IF TTO=0	CHECK AND BRACH IF NO ATTEN
0F0E	AAA2	BSTP 006	BUSY	BSWI 066	RESTRH	BR	NO TYPE OUT WILL OCCUR
0F10	B2F8	BSTP 007	OK	CCCM 173	STORH1	BAL	SAVE H1
0F12	4E86	BSTP 008				H=I	PUT IC IN H REG
0F14	26C3	BSTP 016	ABC			D0=0\$K0C	BUILD
0F16	2707	BSTP 017				D1=0	FLAGS
0F18	2213	BSTP 018				V0=0\$K01	SET UP COUNT
0F1A	96C4	BSTP 019		ALDP 157	ENTRY	BR	GO DISPLAY

 * CROSS REFERENCE FOR CSECT BSTP *

BSTP 002 BSWI 118
 BSTP 006 BSTP 003 BSTP 004
 BSTP 007 BSTP 005

BSWI DESCRIPTIVE TEXT

ENTRY POINTS

- CHECKB - FROM I-CYCLES WHEN EXCEPTIONAL CONDITION, S7=0, IS RECOGNIZED. THE CONTROL WORD AT THIS ENTRY CREATES A VALID NEXT INSTRUCTION ADDRESS BY ENSURING THAT THE HIGH-ORDER BYTE IS CLEAR.
- CHECKE - FROM LOAD PSW ROUTINE AFTER LOADING A PSW OR AFTER AN EFFECTIVE BRANCH INSTRUCTION. IF THE MODE SW IS IN THE INSN STEP POSITION, THE CONTROL WORD AT THIS ENTRY TURNS ON THE INSN STEP LATCH, BB BIT 5. BB5 IS TURNED OFF BY THE START KEY.
- RESTRH - FROM ALTER/DISPLAY AND INTEGRATED I/O ROUTINES. THIS ENTRY RESTORES CPU MODE/ZONE, PROTECT KEY, AND L.S. REGISTERS BEFORE TESTING FOR THE NEXT INTERRUPT OR REQUEST. WHEN THE MICROPROGRAM EXITS FROM BSWI TO HANDLE A REQUEST, A RETURN IS MADE TO THIS ENTRY AFTER THE REQUEST IS PROCESSED.

DESCRIPTION

AFTER ENTRY AT EITHER -CHECKB- OR -CHECKE-, THE TIMER FUNCTION IS HANDLED. WHEN THE TIMER FEATURE IS PRESENT, THE TIMER IN PROGRAM STORAGE IS UPDATED IF THERE IS A VALUE IN THE -TIM- COUNTER. THE -TIM- COUNTER IS THE INVERSE OF THE HARDWARE TIMING COUNTER THAT IS STEPPED BY POWER SUPPLY PULSES. WHEN THE TIMER VALUE IN PROGRAM STORAGE IS REDUCED THROUGH ZERO, AN EXTERNAL INTERRUPT IS INITIATED BY SETTING BC BIT 1.

DESCRIPTION -- CONTINUED

THE FOLLOWING NOT S7 BRANCH CONDITIONS CAUSE ENTRY TO THE BSWI ROUTINE. THESE CONDITIONS ARE LISTED ACCORDING TO PRIORITY OF TESTING. EXCEPT FOR SOFT STOP, THE MICROPROGRAM BRANCHES TO THE APPROPRIATE ROUTINE FOR HANDLING THE CURRENT CONDITION.

- | | |
|---|---------|
| 1. INTEGRATED I/O REQUEST | BB1 = 1 |
| 2. EXTERNAL INTERRUPT MASKED | BB3 = 0 |
| 3. CHANNEL 1 INTERRUPT MASKED | BB6 = 0 |
| 4. COMMUNICATIONS CHANNEL INTERRUPT PENDING | BB7 = 0 |
| 5. MASKED CHANNEL 0 INTERRUPT | BB2 = 0 |
| 6. INSTRUCTION STEP LATCH | BB5 = 1 |
| 7. SET IC LATCH | BB4 = 1 |
| 8. SOFT STOP LATCH | BB0 = 0 |

THE SOFT STOP LATCH IS SET BY ANY ONE OF THE FOLLOWING -- INSN STEP LATCH, SET IC LATCH, ADDR MATCH, STOP SW, OR BY SETTING BCO. IF THE SOFT STOP LATCH IS ON WHEN TESTED, THE MICROPROGRAM LOOPS ON THE TESTS FOR INTEGRATED I/O REQUEST AND SET IC. EXIT FROM THE SOFT STOP LOOP OCCURS WHEN ONE OF THESE TESTS IS SUCCESSFUL OR WHEN THE SOFT STOP LATCH IS TURNED OFF. THE SOFT STOP LATCH IS TURNED OFF BY PRESSING THE START KEY, LOAD KEY, OR CSL KEY.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BSWI 001	T			EXCEPTIONAL CONDITION ROUTINE	D. L. SMITH
		BSWI 002	*			REGISTERS NOT TO BE DESTROYED ARE	
		BSWI 003	*			P0 - CONDITION CODE	
		BSWI 004	*			P1 - PROG MASK AMWP BITS	
		BSWI 005	*			H1 - H1 ORDER 8 BITS OF A 24 BIT BRANCH ADDRESS	
		BSWI 006	*			I01- INSTRUCTION COUNTER	
07C8	2F07	BSWI 007	CHECKB			H1=0	CREATE VALID BRANCH ADDRESS
07CA	3600	BSWI 008	CHECKE			SET BC K=80	CONDITION INSTR STEP
07CC	B2F8	BSWI 009		CCOM 173	STORH1	BAL	STORE H1 IN AUX 00F4
07CE	5AEF	BSWI 011				H0=TIM	READ TIMER VALUE
07D0	1EFF	BSWI 012				H0=H0^KFF	INVERT VALUE
07D2	C4F7	BSWI 013		047	GOINA	BR IF Z=0	BR IF NO VALUE IN TIMER
07D4	5E09	BSWI 020				UO=H0	CALCULATE
07D6	6EE3	BSWI 021				H0=H0+H0	VALUE

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
07DB	6EE3	BSWI 022				HO=HO+HO	FOR 60
07DA	6E03	BSWI 023				HO=HO*U0	CYCLE
07DC	2B55	BSWI 030				T1=0\$K50	READ OUT
07DE	50B8	BSWI 031				RDH U T+2	CURRENT
07E0	52BA	BSWI 032				RDH V T-2	VALUE
07E2	20C2	BSWI 033				SET S3	SET CARRY LATCH
07E4	72E9	BSWI 034				VOC=V0-H0+C	DO
07E6	71A9	BSWI 035				UIC=U1-T0+C	ARITH
07E8	70A9	BSWI 036				UOC=U0-T0+C	FUNCTION ON TIMER
07EA	42AF	BSWI 038				STPO=TO	DISABLE STORAGE PROTECT
07EC	70B8	BSWI 040				STH U T+2	STORE
07EE	72BA	BSWI 041				STH V T-2	RESULT
07F0	A7FA	BSWI 043		CCOM 181	RSTRKY	BAL	RESTORE CPU KEY
07F2	F5F9	BSWI 045		048	CKWAIT	BR IF S3=1	BR IF NOT THRU 0
07F4	2608	BSWI 046				SET BC K=40	TURN ON TIMER INT
07F6	AA84	BSWI 047	GOINA	051	INA	BR	MAKE ADDRESSABLE
07F8	E976	BSWI 048	CKWAIT	047	GOINA	BR IF P1 BIT6=0	BR IF NOT IN WAIT STATE
07FA	9CC6	BSWI 049		BPSW 062	WAIT	BR	TO AVOID RUNNING METER IN WAIT
2A84	DFE8	BSWI 051	INA	078	INB	BR IF BB1=0	BR IF NO NATIVE REQUEST
2A86	2040	BSWI 052				SET S5	
2A88	2080	BSWI 053	INH			SET S4	
2A8A	2406	BSWI 054	ISTRDR			SET MODE K=30	**SET MODE=110 FOR RDR
2A8C	E990	BSWI 055		057	TSTPCH	BR IF RS BIT6=0	BR IF NO RDDR REQ
2A8E	9EBA	BSWI 056		ERRQ 029	RDST	BR	**GO TO RDR ROUTINE
2A90	2402	BSWI 057	TSTPCH			SET MODE K=10	SET MODE=010 FOR PCH
2A92	EB96	BSWI 058		060	TSTTWR	BR IF PS BIT6=0	GO TO PCH ROUTINE
2A94	5EEA	BSWI 059		ERRQ 016	PCHST	BR	**GO TO PCH ROUTINE
2A96	2404	BSWI 060	TSTTWR			SET MODE K=20	**SET MODE=100 FOR 1052
2A98	EA9C	BSWI 061		063	TSTPTR	BR IF TT BIT6=0	**BR IF NO 1052 REQUEST
2A9A	9F10	BSWI 062		DYPE 002	TWRREQ	BR	**GO TO 1052 ROUTINE
2A9C	2482	BSWI 063	TSTPTR			SET MODE K=18	SET 1403 PRINTER MODE
2A9E	FAA2	BSWI 064		066	RESTRH	BR IF PRS BIT7=0	BR IF NO PTR REQ
2AA0	9D6A	BSWI 065		DPTQ 006	PTRREQ	BR	
2AA2	2400	BSWI 066	RESTRH			SET MODE K=00	SET CPU MODE AND ZONE
2AA4	2A07	BSWI 067				TO=0	RST TO FOR NATIVES
2AA6	A7FA	BSWI 069		CCOM 181	RSTRKY	BAL	RESTORE CPU KEY
2AA8	E1AD	BSWI 071		073	NOTPI	BR IF S6=1	BR IF P, I NOT STORED
2AAA	9258	BSWI 072		CCOM 066	LRSTRB	BAL	RESTORE P, I
2AAC	C1E3	BSWI 073	NOTPI	075	NOTSIO	BR IF S4=1	BR IF FROM REQ AND NOT SIO
2AAE	8183	BSWI 074		CICY 005	ISTART N	N=S BITS67	RTN TO I CYCLES
2AE2	B354	BSWI 075	NOTSIO	CCOM 186	RESTRH	BAL	RESTORE CPU BYTE H1
2AE4	D1BE	BSWI 076		105	IND	BR IF S5=0	
2AE6	C040	BSWI 077				RST S5	
2AE8	FFB1	BSWI 078	INB	091	INC	BR IF BB3=1	BR IF NO EXTERNAL INT.
2AEA	59BF	BSWI 082				T1=XINT	READ INTERRUPT REG FOR INT CODE
2AEC	1BFF	BSWI 083				T1=T1\$KFF	INVERT IT
2AEE	0640	BSWI 088				RST BC K=04	
2AF0	2115	BSWI 089				U1=0\$K10	INTERRUPT
2AF2	9C74	BSWI 090		BPSW 005	ENTRYB	BR	FUNCTION
2AB0	EBB5	BSWI 091	INC	099	INE	BR IF BB6=1	BR IF NO CHAN 1 INTR.
2AB2	A2FA	BSWI 093		DCLT 047	INTENT	BR	GO TO CHANNEL 1
2AB4	FBB9	BSWI 099	INE	101	INF	BR IF BB7=1	BR IF NO COMMUN. CHNL INTERRUPT
2AB6	5007	BSWI 100		GINT*****	INTCOM	BR	GO TO COMMUNICATIONS

BSYS DESCRIPTIVE TEXT

THE SYSTEM RESET ROUTINE IS ENTERED AT LABEL * SYSRST * FROM THE CHECK SUM ROUTINE -BCHK-, TO INITIALIZE CERTAIN I/O LATCHES AND AUXILIARY STORAGE LOCATIONS.

THIS ENTRY IS MADE AFTER EITHER THE SYSTEM RESET, LOAD, OR CONTROL STORAGE LOAD KEYS ARE PRESSED.

WHEN THE LOAD KEY IS PRESSED, THE FOLLOWING STEPS TAKE PLACE PRIOR TO ENTRY AT LABEL * SYSRST * -

- 1 A HARDWARE RESET IS PERFORMED.
- 2 THE RESIDENT DIAGNOSTIC -BDIA- IS EXECUTED.
- 3 THE CHECK SUM ROUTINE -BCHK- IS EXECUTED.
- 4 -BSYS- IS ENTERED AT LABEL * SYSRST *.
- 5 THE -DCLL- ROUTINE IS BRANCHED TO FOR IPL START.
- 6 WHEN IPL IS COMPLETED, A BRANCH TO -BSYS- AT LABEL * RETURN *.
- 7 BRANCH TO -BPSW- TO LOAD THE INITIAL PSW.

WHEN THE SYSTEM RESET KEY IS PRESSED, THE FOLLOWING STEPS TAKE PLACE PRIOR TO ENTRY AT LABEL * SYSRST * -

- 1 A HARDWARE RESET IS PERFORMED.
- 2 THE RESIDENT DIAGNOSTIC -BDIA- IS EXECUTED.
- 3 THE CHECK SUM ROUTINE -BCHK- IS EXECUTED.
- 4 -BSYS- IS ENTERED AT LABEL * SYSRST *.

WHEN THE CSL KEY IS PRESSED, THE FOLLOWING STEPS TAKE PLACE PRIOR TO ENTRY AT LABEL * SYSRST * -

- 1 A HARDWARE RESET IS PERFORMED.
- 2 THE -BCPL- ROUTINE IS EXECUTED.
- 3 THE RESIDENT DIAGNOSTIC -BDIA- IS EXECUTED.
- 4 THE CHECK SUM ROUTINE -BCHK- IS EXECUTED.
- 5 -BSYS- IS ENTERED AT LABEL * SYSRST *.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BSYS 001	T			SYSTEM RESET	
216C	3210	BSYS 002	SYSRST			SET MMSK K=81	SET SYS RST PRIORITY,BLOCK TRAPS
216E	2460	BSYS 004				SET MODE K=06	SET 2540 ZONE
2170	2607	BSYS 005				DO=0	RST 2540 FLAG REG
2172	24F6	BSYS 018				SET MODE K=3F	SET CHNL MODE AND ZONE
2174	4052	BSYS 019				RDH U DC,9A	RST U
2176	4206	BSYS 020				V=U	
2178	4406	BSYS 021				G=U	
217A	4606	BSYS 022				D=U	
217C	2D08	BSYS 026				SET GB K=40	RST MULTIPLEX LATCH
217E	2490	BSYS 028				SET MODE K=09	SET TO 2311 MODE AND ZONE
2180	AFE8	BSYS 029	GOBAL	FILE 280	SETDA	BAL	SET DEVICE ADDR & SELECT MODULE
2182	8224	BSYS 030		FILE 319	RSTATT	BAL	GO RST GATED ATTENTION
2184	49CF	BSYS 031				MS=PO	RESET MODULE SELECT
2186	E581	BSYS 032		029	GOBAL	BR IF DAS12=1	BR IF ANY MORE GATED ATTENTION
2188	2EC8	BSYS 033				SET FIB K=40	RST FILE FLAG REG.
218A	2482	BSYS 037				SET MODE K=18	SET 1403 MODE
218C	4A52	BSYS 039				RDH T DC,9A	SET T=0000
218E	4E42	BSYS 040				RDH H DC,98	SET H=00C8
2190	5EE6	BSYS 042				H=H-1	H=C0C7
2192	57E8	BSYS 043				RDB D1 AS,H+1	GET PCCL CONSTANT
2194	457F	BSYS 044				PCCL=D1	SET PCCL LATCH FOR 1403
2196	2400	BSYS 045				SET MODE K=00	SET CPU MODE AND ZONE
2198	6EE4	BSYS 046				H=H+2	H=00CA

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
219A	56E0	BSYS 047				RDH D AS,H+0	GET UNIT STATUS AND SENSE BYTE
219C	26C7	BSYS 048				DO=0	RST UNIT STATUS
219E	577D	BSYS 049				D1=D1L	RST SENSE BYTE LOW
21A0	17B3	BSYS 050				D1=D1*-K0B	REGEN BLOCK DATA CHECK
21A2	76EA	BSYS 051				STH D AS,H-2	CLEAR 1403 SENSE
21A4	7AE0	BSYS 053	LOOP1			STH T AS,H+0	ZERO OUT 1ST HALFWORD OF THE
21A6	2F1D	BSYS 054				H1=H1+K10	NATIVE DEVICES' UCW
21A8	F4A4	BSYS 055		053	LOOP1	BR IF AC=0	(IN 00C8, 00D8, 00E8, 00F8)
21AA	2F91	BSYS 056				H1=0-K09	H=COF6
21AC	7AEA	BSYS 057				STH T AS,H-2	CLEAR 1052 SENSE BYTE
21AE	2E15	BSYS 059				H0=0\$K10	H=10F4
21B0	7AEA	BSYS 060				STH T AS,H-2	RST PCH INDICATORS & SENSE BYTE
21B2	2FF5	BSYS 061				H1=0\$KFO	H=10F0
21B4	7AE0	BSYS 062				STH T AS,H+0	RST RDR INDICATORS & SENSE BYTE
21B6	7A32	BSYS 064				STH T DA,8E	**RST STATUS/ACT BYTE FOR CHNL 1 OR 2311
21B8	7A52	BSYS 066				STH T DA,9A	RST 2311 SENSE HALF WORD 1
21BA	7A62	BSYS 067				STH T DA,9C	RST 2311 SENSE HALF WORD 2
21BC	40A6	BSYS 118				U=T	
21BE	3212	BSYS 119				SET MMSK K=91	SET MCH CK TRAP BIT, STOP IF ERR
21C0	1210	BSYS 120				RST MMSK K=81	RST SYS RST BIT TO ALLOW TRAPS
21C2	20A0	BSYS 121				SET S K=0A	
21C4	C3C8	BSYS 122		124	SYSTEM	BR IF BA4=0	BR IF NOT IPL
21C6	A5DA	BSYS 123		DCLL 002	IPL	BR	
21C8	E8D2	BSYS 124	SYSTEM	129	RESPSW	BR IF DR6=0	BR IF NOT PSW RESTART
21CA	C62C	BSYS 125	RETURN			RST BC K=62	
21CC	1212	BSYS 126				RST MMSK K=91	
21CE	4052	BSYS 127				RDH U DC,9A	ZERO OUT U REG
21D0	5C9A	BSYS 128		BPSW 024	LPSW	BR	
21D2	162C	BSYS 129	RESPSW			RST BC K=E2	SET STOP,RST CSL,RST SYS,RST IPL
21D4	1212	BSYS 130				RST MMSK K=91	RST TRAP
21D6	5C82	BSYS 131				RDH P DA,A8	LOAD
21D8	44CF	BSYS 132				SM=PO	UP
21DA	42DF	BSYS 133				STPO=P1	THE
21DC	4852	BSYS 134				RDH I DC,9A	LAST
21DE	5CC2	BSYS 135				RDH P DA,B8	PSW
21E0	2FG7	BSYS 136				H1=0	ZERO HIGH ORDER ADDR
21E2	5CB2	BSYS 137		BPSW 038	NOT	BR	
00B4	0E0C	BSYS 138	ATABLE	ADDR=00B4			
		BSYS 148	C			XAX'0E0C0D1F'	
		BSYS 149	AEND				
00F4	0090	BSYS 151	ATABLE	ADDR=00F4			
		BSYS 155	C			XAX'0090'	
		AEND					

 * CROSS REFERENCE FOR CSECT BSYS *

BSYS 002 BCHK 082
 BSYS 029 BSYS 032
 BSYS 053 BSYS 055
 BSYS 124 BSYS 122
 BSYS 125 DCLL 041
 BSYS 129 BSYS 124

BWRP DESCRIPTIVE TEXT

ENTRY POINT

THIS ROUTINE IS ENTERED WHEN A TRAP ADDRESS OF 0210 IS FORCED BY --

1. STORAGE PROTECT VIOLATION. PROTECT KEYS DO NOT MATCH.
2. STORAGE WRAP VIOLATION. THIS OCCURS WHEN A PROGRAM STORAGE ADDRESS THAT EXCEEDS THE SYSTEM SIZE IS ENCOUNTERED.

SYSTEM SIZE	PROG	STOR	ADDR	RANGE
16K		0000	-	3FFF
24K		0000	-	5FFF
32K		0000	-	7FFF
48K		0000	-	BFFF

DESCRIPTION

THE UNCONDITIONAL BRANCH WORD AT THE START OF THE ROUTINE STORES THE ADDRESS +2 OF THE TRAPPED WORD IN BITS 2-6 OF THE I-REG BACKUP, I.E., ZONE 4. BIT 7 OF THE I-REG BACKUP IS SET FROM DYN BIT 3. IF THE TRAP OCCURS DURING INSTRUCTION READOUT IN I-CYCLES, THE ADDRESS STORED IN THE BACKUP I-REG IS THE ADDRESS OF THE TRAPPED WORD +3 BECAUSE DYN BIT 3 IS SET TO 1 LATER IN THE ROUTINE, THIS BACKUP ADDRESS IS DECREMENTED BY 1 TO POINT TO THE CONTROL WORD THAT FOLLOWS THE TRAPPED WORD. THIS CONTROL WORD IS USED TO DETERMINE WHICH HALFWORD OF THE INSTRUCTION WAS BEING READ WHEN THE TRAP OCCURRED. THIS IS DONE IN ORDER TO SET UP THE NECESSARY REGISTERS FOR THE PENDING INTERRUPT.

AFTER THE UNCONDITIONAL BRANCH, DYN BIT 1 IS TESTED. THIS BIT IS SET TO 1 FOR A STORAGE WRAP VIOLATION. IF DYN1 = 0, A STORAGE PROTECT VIOLATION HAS OCCURRED. FOR EITHER VIOLATION, THE ROUTINE DETERMINES THE MODE BY TESTING BA BITS 1,2, AND 3. FOR FILE, COMMUNICATIONS, OR CHANNEL MODES, THE MICROPROGRAM BRANCHES TO THE APPROPRIATE ROUTINE TO ATTEMPT THE INTERRUPT. FOR THE INTEGRATED I/O UNITS REMAINING, EITHER PROTECTION CHECK OR PROGRAM CHECK IS SET IN THE CHANNEL STATUS OF THE APPROPRIATE UCW, THEN THE INTERRUPT IS ATTEMPTED.

WHEN A STORAGE PROTECT VIOLATION IN CPU MODE OCCURS, THE PROTECTION ERROR IS FLAGGED, AND THE PROGRAM INTERRUPT IS INITIATED.

WHEN A STORAGE WRAP VIOLATION IN CPU MODE OCCURS, TESTS ARE MADE TO DETERMINE IF THE WRAP OCCURRED DURING INSTRUCTION READOUT, AS PREVIOUSLY STATED. FOR ANY STORAGE WRAP VIOLATION IN CPU MODE, THE ROUTINE EXITS TO FLAG THE ADDRESS ERROR AND INITIATE THE INTERRUPT.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BWRP 001	T			WRAP TRAP AND STORAGE	D. L. SMITH
		BWRP 002	*				
		BWRP 003	*				
		BWRP 004	*				
		BWRP 005	*				
		BWRP 006	*				
		BWRP 007	ATABLE	ADDR=0210			
0210	E87C	BWRP 008		010	NXTADR	BR	SAVE BACKUP ADDR IF IN CPU MODE
		BWRP 009	AEND				
087C	D4FA	BWRP 010	NXTADR	020	PROTCK	BR IF DYN1=0	BR IF STORAGE PROTECT
087E	9795	BWRP 011		012	STRG N	N=BA BITS123	FIND THE MODE CAUSING WRAP TRAP
0A80	F790	BWRP 012	STRG 0	037	ADDRCK	BR IF BA3=0	
0A82	94B4	BWRP 013	STRG 1	FILT 079	WRAP	BR	2311 MODE
0A84	B35E	BWRP 014	STRG 2	090	PUNRP	BR	2540 PUNCH WRAP
0A86	9270	BWRP 015	STRG 3	077	WRAP	BR	1403 MODE
0A88	9270	BWRP 016	STRG 4	077	WRAP	BR	1052 MODE

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0A8A	5007	BWRP 017	STRG 5	GAN3*****	COMWRP	BR	COMMUNICATIONS
0A8C	926E	BWRP 018	STRG 6	076	RDRPU	BR	2540 READER WRAP
0A8E	E476	BWRP 019	STRG 7	DCLR 106	WRAP	BR	CHANNEL
0A7A	97E5	BWRP 020	PROTCK	021	PRT N	N=BA BITS123	FIND THE MODE CAUSING THE PROT.
0A60	9018	BWRP 021	PRT 0	066	CPUPRT	BR	CPU MODE
0A62	9480	BWRP 022	PRT 1	FILT 077	PRTCHK	BR	2311 MODE
0A64	9268	BWRP 023	PRT 2	073	RDPUR	BR	2540 PUNCH PROTECT CK
0A60	926A	BWRP 024	PRT 3	074	PROTEC	BR	1403 MODE
0A68	926A	BWRP 025	PRT 4	074	PROTEC	BR	1052 MODE
0A6A	5007	BWRP 026	PRT 5	GAN3*****	PSUEDD	BR	COMMUNICATIONS
0A6C	9268	BWRP 027	PRT 6	073	RDPUR	BR	2540 READER PROTECT CK
0A6E	9614	BWRP 028	PRT 7	DCLR 108	PROTCH	BR	CHANNEL
0A90	E1AE	BWRP 037	ADDRCK	064	NXTD	BR IF S6=0	IF EXECUTE ON ADDR. OK
0A92	08CD	BWRP 040				Z=I0KCO	CHECK FAILURE
0A94	C4AE	BWRP 048		064	NXTD	BR IF ZNZ	DURING INSTRUCTION
0A96	092B	BWRP 049				Z=I1K02	READOUT
0A98	C4AE	BWRP 050		064	NXTD	BR IF ZNZ	
0A9A	2440	BWRP 051				SET MODE K=04	PUT IN BACKUP ZONE
0A9C	78F2	BWRP 052				STH I DA, BE	SAVE ADDRESS STORED
0A9E	2400	BWRP 053				SET MODE K=00	PUT BACK IN CPU ZONE
0AA0	56F2	BWRP 054				RDH D DA, BE	GET ADDRESS IN TO WORK ON
0AA2	16C5	BWRP 055				DO=DO*-KCO	
0AA4	372B	BWRP 056				D1=D1-K02	
0AA6	466C	BWRP 057				RDH D CS, D	READ CONTROL WORD FOLLOWING
0AA8	D633	BWRP 058		061	NXTA	BR IF D01=1	1ST HALFWORD TRAPPED BR NO
0AAA	24C7	BWRP 059	NXTC			GO=0	RESET ILC CODE
0AAC	A9C0	BWRP 060	NXTB	CICY 045	ADDERR 0	BR	
0AB2	E22D	BWRP 061	NXTA	060	NXTB	BR IF D06=1	BR IF THIRD READOUT FAILED
0AB4	6886	BWRP 062				I=1-2	BUILD IT AS AN
0AB6	8AAA	BWRP 063		059	NXTC	BR	RR OP FLAGGED
0AAE	2AC7	BWRP 064	NXTD			T0=0	
0AB0	A9C0	BWRP 065		CICY 045	ADDERR 0	BR	
1018	2AC7	BWRP 066	CPUPRT			T0=0	FLAG
101A	2B43	BWRP 067				T1=0\$K04	PROTECTION
101C	2F07	BWRP 068				H1=0	ERROR
101E	9C72	BWRP 069		BPSW 004	PRGGA	BR	
		BWRP 070	*				*****
		BWRP 071	*				NATIVE I/O WRAP AND PROTECT
		BWRP 072	*				*****
1268	26C7	BWRP 073	RDPUR			DO=0	FIX DO FO 2540
126A	2E95	BWRP 074	PROTEC			H0=0\$K90	SET PROTECT AND SEC.
126C	C4F2	BWRP 075		078	WRCPRP	BR IF ZNZ	UNC BR
126E	26C7	BWRP 076	RDRPU			DO=0	FIX DO FO 2540
1270	2EA5	BWRP 077	WRAP			H0=0\$KAO	SET PRGM. CHECK AND SEC.
1272	1773	BWRP 078	WRCPRP			D1=D1*-K07	D1=XXX8
1274	5460	BWRP 079				RDH G AS, D+0	READ CHANNEL STATUS
1276	64E5	BWRP 080				GO=GO\$HO	OR IN PRGM. CHECK OR PROT. CHK
1278	2543	BWRP 081				G1=0\$K04	RESET CHAINING--KEEP ACTIVE ON
127A	2EC3	BWRP 082				H0=0\$KOC	SET CH END DEV END UNIT STATUS
127C	7468	BWRP 083				STH G AS, D+2	D=XXX8 CHAN STATUS AND FLAGS
127E	7E6A	BWRP 084				STH H AS, D-2	D=XXXA UNIT STATUS
1280	E765	BWRP 085		086	WHICHN N	N=D1 BITS23	BREAK OUT NATIVES
1260	9D52	BWRP 086	WHICHN 0	DPTQ 019	TAKEIN	BR	1403 * TRY

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1262	BC9E	BWRP 087	WHICHN 1	DCLR 038	NTYPE	BR	2540 READER * TO
1264	BC9E	BWRP 088	WHICHN 2	DCLR 038	NTYPE	BR	2540 PUNCH * CAUSE INT
1266	9F20	BWRP 089	WHICHN 3	DYPE 010	NOATEN	BR	1052 CAUSE INTERRUPT
335E	A062	BWRP 090	PUNRP	ERRQ 087	RPHUCW	BAL	SET D TO 00E8
3360	9270	BWRP 091		077	WRAP	BR	

 * CROSS REFERENCE FOR CSECT BWRP *

BWRP 010	BWRP 0C8		
BWRP 012	BWRP 011		
BWRP 020	BWRP 010		
BWRP 021	BWRP 020		
BWRP 037	BWRP 012		
BWRP 059	BWRP 0E3		
BWRP 060	BWRP 0E1		
BWRP 061	BWRP 058		
BWRP 064	BWRP 037	BWRP 048	BWRP 050
BWRP 066	BWRP 021		
BWRP 073	BWRP 023	BWRP 027	
BWRP 074	BWRP 024	BWRP 025	
BWRP 076	BWRP 018		
BWRP 077	BWRP 015	BWRP 016	BWRP 091
BWRP 078	BWRP 075		
BWRP 086	BWRP 0E5		
BWRP 090	BWRP 014		

CBIN DESCRIPTION

ENTRY	OPERATION	INSTRUCTION FORMAT			
		BYTE 1	BYTE 2	BYTE 3	BYTE 4
RR0P19	COMPARE	19	R1 R2	.	.
RR0P1A	ADD	1A	R1 R2	.	.
RR0P1B	SUBTRACT	1B	R1 R2	.	.
RR0P1E	ADD LOGICAL	1E	R1 R2	.	.
RR0P1F	SUBTRACT LOGICAL	1F	R1 R2	.	.
RX0P49	COMPARE HALFWORD	49	R1 X2	B2 D2	D2
RX0P4A	ADD HALFWORD	4A	R1 X2	B2 D2	D2
RX0P4B	SUBTRACT HALFWORD	4B	R1 X2	B2 D2	D2
RX0P59	COMPARE	59	R1 X2	B2 D2	D2
RX0P5A	ADD	5A	R1 X2	B2 D2	D2
RX0P5B	SUBTRACT	5B	R1 X2	B2 D2	D2
RX0P5E	ADD LOGICAL	5E	R1 X2	B2 D2	D2
RX0P5F	SUBTRACT LOGICAL	5F	R1 X2	B2 D2	D2

OBJECTIVES

- ADD - THE SECOND OPERAND IS ADDED TO THE FIRST OPERAND, AND THE SUM IS PLACED IN THE FIRST OPERAND LOCATION.
- ADD LOGICAL - SAME AS ADD. THE CONDITION CODES DIFFER.
- SUBTRACT - THE SECOND OPERAND IS SUBTRACTED FROM THE FIRST OPERAND, AND THE DIFFERENCE IS PLACED IN THE FIRST OPERAND LOCATION.
- SUBT. LOG. - SAME AS SUBT. THE CONDITION CODES DIFFER
- COMPARE - THE FIRST OPERAND IS COMPARED WITH THE SECOND OPERAND, AND THE RESULT DETERMINES THE CONDITION CODE. THE COMPARE IS DONE BY COMPLEMENT ADDING THE OPERANDS.

GENERAL OBJECTIVES

1. FETCH OPERANDS
2. ADD OR COMPLEMENT ADD OPERANDS
3. SET CONDITION CODE
4. RETURN TO I-CYCLES

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CBIN 001	T			BINARY ADD AND SUB COMP	
		CBIN 002	*				
		CBIN 003	*				
		CBIN 004	*				** RR COMPARE
1148	8A70	CBIN 005	RR0P19	CCOM 036	CRGET	BAL	FETCH OPERANDS
114A	A8CA	CBIN 006		009	CXCOMP	BR	GO COMPLEMENT ADD OPERANDS
		CBIN 007	*				** RX COMPARE
28C8	A770	CBIN 008	RX0P59	CCOM 045	CXGET	BAL	FETCH OPERANDS
28CA	8174	CBIN 009	CXCOMP	CCOM 086	CSLOOP	BAL	COMPLEMENT ADD OPERANDS
28CC	E4D0	CBIN 010		012	OVER	BR IF NOVFL	BR IF NO OVERFLOW
28CE	168D	CBIN 011				DO=DO=K80	INVERT SIGN
28D0	E5D7	CBIN 012	OVER	015	NZO	BR IF S2=1	BR IF RESULT NOT ZERO
28D2	2CG7	CBIN 013				PO=0	SET CONDITION CODE 0 -- EQUAL
28D4	81B3	CBIN 014		CICY 005	ISTART N	N=S BITS67	RETURN TO I-CYCLES
28D6	C645	CBIN 015	NZO	018	ALOW	BR IF D00=1	BR IF RESULT IS MINUS
28D8	2C25	CBIN 016				PO=0\$K20	SET CONDITION CODE 2 -- A-HIGH
28DA	81B3	CBIN 017		CICY 005	ISTART N	N=S BITS67	RETURN TO I-CYCLES
28C4	2C55	CBIN 018	ALOW			PO=0\$K50	SET CONDITION CODE 1 -- A-LOW
28C6	81B3	CBIN 019		CICY 005	ISTART N	N=S BITS67	RETURN TO I-CYCLES
		CBIN 020	*				
		CBIN 021	*				** RX COMPARE HALFWORD
3388	ADA8	CBIN 022	RX0P49	CCOM 012	CXGETH	BAL	FETCH OPERANDS, EXTEND SIGN

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
338A	A8CA	CBIN 023		009	CXCGMP	BR	GO COMPLEMENT ADD OPERANDS
		CBIN 024	*				
		CBIN 025	*				** RR ADD
3366	8A70	CBIN 026	RROPIA	CCOM 036	CRGET	BAL	FETCH OPERANDS
3368	AC1A	CBIN 027		031	CXADD	BR	GO ADD OPERANDS
		CBIN 028	*				
		CBIN 029	*				** RX ADD
2C18	A770	CBIN 030	RXOP5A	CCOM 045	CXGET	BAL	FETCH OPERANDS
2C1A	8176	CBIN 031	CXADD	CCOM 087	CXLOOP	BAL	ADD OPERANDS
2C1C	E4BD	CBIN 032	CAC1BC	CLST 009	OVCHEK	BR IF OVFL	BR IF OVERFLOW TO SET CC3
2C1E	E5B0	CBIN 033		CLST 034	NEGCC0	BR IF S2=0	BR IF ZERO RESULT TO SET CC0
2C20	C625	CBIN 034	CCCHK	CLST 036	NEGCC1	BR IF D00=1	BR IF NEG SIGN TO SET CC1
2C22	AC46	CBIN 035		CLST 024	TESCC2	BR	BR TO SET CONDITION CODE 2
		CBIN 036	*				
		CBIN 037	*				** RX ADD HALFWORD
1A7C	ADA8	CBIN 038	RXOP4A	CCOM 012	CXGETH	BAL	FETCH OPERANDS
1A7E	AC1A	CBIN 039		031	CXADD	BR	GO ADD OPERANDS
		CBIN 040	*				
		CBIN 041	*				** RR SUBTRACT
01F4	8A70	CBIN 042	RROPIB	CCOM 036	CRGET	BAL	FETCH OPERANDS
01F6	810C	CBIN 043		047	CXSUB	BR	GO COMPLEMENT ADD OPERANDS
		CBIN 044	*				
		CBIN 045	*				** RX SUBTRACT
010A	A770	CBIN 046	RXOP5B	CCOM 045	CXGET	BAL	FETCH OPERANDS
010C	8174	CBIN 047	CXSUB	CCOM 086	CSLOOP	BAL	COMPLEMENT ADD OPERANDS
010E	AC1C	CBIN 048		032	CAC1BC	BR	GO TEST FOR CONDITION CODE
		CBIN 049	*				
		CBIN 050	*				** RX SUBTRACT HALFWORD
3176	ADA8	CBIN 051	RXOP4B	CCOM 012	CXGETH	BAL	FETCH OPERANDS
3178	810C	CBIN 052		047	CXSUB	BR	GO COMPLEMENT ADD OPERANDS
		CBIN 053	*				
		CBIN 054	*				** RR ADD LOGICAL
3380	8A70	CBIN 055	RROPIE	CCOM 036	CRGET	BAL	FETCH OPERANDS
3382	ADC8	CBIN 056		060	CXADDL	BR	GO ADD OPERANDS
		CBIN 057	*				
		CBIN 058	*				** RX ADD LOGICAL
2DC6	A770	CBIN 059	RXOP5E	CCOM 045	CXGET	BAL	FETCH OPERANDS
2DC8	8176	CBIN 060	CXADDL	CCOM 087	CXLOOP	BAL	GO ADD OPERANDS
2DCA	76A8	CBIN 061	S2S3CK			STH D AS,T+2	PUT RESULT IN
2DCC	7EAA	CBIN 062				STH H AS,T-2	FIRST OPERAND LOCATION
2DCE	E5DB	CBIN 063		067	NONZER	BR IF S2=1	BR IF RESULT NOT ZERO
2DD0	F507	CBIN 064		072	ZRCARY	BR IF S3=1	BR IF 0-BIT CARRY
2DD2	2C07	CBIN 065				PO=0	SET CONDITION CODE 0
2DD4	81B3	CBIN 066		CICY 005	ISTART N	N=S BITS67	RETURN TO I-CYCLES
2DDA	F5C3	CBIN 067	NONZER	070	NZCARY	BR IF S3=1	BR IF ZERO-BIT CARRY
2DDC	2C55	CBIN 068				PO=0\$K50	SET CONDITION CODE 1
2DDE	81B3	CBIN 069		CICY 005	ISTART N	N=S BITS67	RETURN TO I-CYCLES
2DC2	2C75	CBIN 070	NZCARY			PO=0\$K70	SET CONDITION CODE 3
2DC4	81B3	CBIN 071		CICY 005	ISTART N	N=S BITS67	RETURN TO I-CYCLES
2DD6	2C25	CBIN 072	ZRCARY			PO=0\$K20	SET CONDITION CODE 2
2DD8	81B3	CBIN 073		CICY 005	ISTART N	N=S BITS67	RETURN TO I-CYCLES
		CBIN 074	*				
		CBIN 075	*				** RR SUBTRACT LOGICAL

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
336E	8A70	CBIN 076	RROPIF	CCOM 036	CRGET	BAL	FETCH OPERANDS
3370	AF74	CBIN 077		081	CXSUBL	BR	GO COMPLEMENT ADD OPERANDS
		CBIN 078	*				
		CBIN 079	*				** RX SUBTRACT LOGICAL
2F72	A770	CBIN 080	RXOP5F	CCOM 045	CXGET	BAL	FETCH OPERANDS
2F74	8174	CBIN 081	CXSUBL	CCOM 086	CSLOOP	BAL	COMPLEMENT ADD OPERANDS
2F76	ADCA	CBIN 082		061	S2S3CK	BR	GO STORE RESULT & SET COND CODE

 * CROSS REFERENCE FOR CSECT CBIN *

CBIN 005	CICY 156	
CBIN 008	CICY 188	
CBIN 009	CBIN 006	CBIN 023
CBIN 012	CBIN 010	
CBIN 015	CBIN 012	
CBIN 018	CBIN 015	
CBIN 022	CICY 172	
CBIN 026	CICY 157	
CBIN 030	CICY 189	
CBIN 031	CBIN 027	CBIN 039
CBIN 032	CBIN 048	
CBIN 038	CICY 173	
CBIN 042	CICY 158	
CBIN 046	CICY 150	
CBIN 047	CBIN 043	CBIN 052
CBIN 051	CICY 174	
CBIN 055	CICY 161	
CBIN 059	CICY 193	
CBIN 060	CBIN 056	
CBIN 061	CBIN 082	
CBIN 067	CBIN 063	
CBIN 070	CBIN 067	
CBIN 072	CBIN 064	
CBIN 076	CICY 162	
CBIN 080	CICY 194	
CBIN 081	CBIN 077	

CBRC DESCRIPTION
BRANCH INSTRUCTIONS

ENTRY	OPERATION	BYTE 1	BYTE 2	BYTE 3	BYTE 4
RRPO05	BRANCH AND LINK	05	R1 R2		
RRPO06	BRANCH ON COUNT	06	R1 R2		
RRPO07	BR ON CONDITION	07	M1 R2		
RXOP44	EXECUTE	44	R1 X2	B2 D2	D2
RXOP45	BRANCH AND LINK	45	R1 X2	B2 D2	D2
RXOP46	BRANCH ON COUNT	46	R1 X2	B2 D2	D2
RXOP47	BR ON CONDITION	47	M1 X2	B2 D2	D2
RSOP86	BR ON INDEX HIGH	86	R1 R3	B2 D2	D2
RSOP86	BRANCH ON INDEX LOW OR EQUAL	87	R1 R3	B2 D2	D2

OBJECTIVES CONTINUED

BR ON CONDITION - IF CONDITION CODE AND MASK MATCH, BRANCH TO DEVELOPED ADDRESS.
NOTE - THE CONDITION CODE IN LS REG PO IS CODED AS FOLLOWS.

COND CODE	PO HIGH
0	0
1	5
2	2
3	7

EXECUTE - BITS 8-15 OF THE SUBJECT INSTRUCTION ARE MODIFIED BY OR-ING THEM WITH BYTE 3 OF R1. THEN THE MODIFIED INSTRUCTION IS EXECUTED. THE SUBJECT INSTRUCTION IS LOCATED AT THE 2ND OPERAND EFFECTIVE ADDRESS.

OBJECTIVES

BRANCH AND LINK - STORE THE UPDATED RIGHTMOST 32 BITS OF THE CURRENT PSW, AND SET THE BRANCH ADDRESS INTO THE INSTRUCTION COUNTER.

BRANCH ON COUNT - BRANCH TO DEVELOPED ADDRESS, EACH TIME, UNTIL SUM IN R1 EQUALS ZERO. EACH TIME THROUGH, SUBTRACT 1 FROM VALUE IN R1.

BR ON INDEX HIGH- ADD THE CONTENTS OF R3 TO R1. COMPARE THIS SUM TO THE CONTENTS OF R3, OR R3 PLUS 1 IF R3 IS EVEN. IF COMPARE IS HIGH, BRANCH TO THE EFFECTIVE SECOND OPERAND ADDRESS.

BRANCH ON INDEX LOW OR EQUAL - SAME AS BRANCH ON INDEX HIGH, EXCEPT THE BRANCH IS ON LOW OR EQUAL COMPARE.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CBRC 001	T			BRANCH INSTRUCTIONS	D. L. SMITH
		CBRC 002	*				
		CBRC 003	*				
		CBRC 004	*			BRANCH CONDITION RR	
28BA	55B3	CBRC 005	RRPO07			T1=G1XH	EXECUTION
28BC	C483	CBRC 006		013	LEG 1	BR IF Z=0	OF
28BE	5EA8	CBRC 007				RDH H AS,T+2	RR
28C0	50A0	CBRC 008				RDH U AS,T+0	BCR
		CBRC 009	*				
		CBRC 010	*			BRANCH CONDITION RX	
28C2	9C11	CBRC 011	RXOP47	012	LEG N	N=PO BITS123	EXECUTION
2880	C5C7	CBRC 012	LEG 0	015	LEG 3	BR IF G10=1	OF
2882	81B3	CBRC 013	LEG 1	CICY 005	ISTART N	N=S BITS67	RX AND
2884	E502	CBRC 014	LEG 2	013	LEG 1	BR IF G12=0	RR
2886	4806	CBRC 015	LEG 3			I=U	BRANCH
2888	9CBA	CBRC 016	LEG 4	BPSW 053	INTBR	BR	CONDITION

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
288A	D5C7	CBRC 017	LEG 5	015	LEG 3	BR IF G11=1	
288C	81B3	CBRC 018	LEG 6	CICY 005	ISTART N	N=S BITS67	
288E	F507	CBRC 019	LEG 7	015	LEG 3	BR IF G13=1	
2890	81B3	CBRC 020		CICY 005	ISTART N	N=S BITS67	
		CBRC 021	*				
		CBRC 022	*		BRANCH AND LINK	RR	
2892	0040	CBRC 023	RROP05			RST S5	EXECUTION
2894	55B3	CBRC 024				T1=G1XH	RR
2896	C49F	CBRC 025		032	OP45A	BR IF Z=0	BALR
2898	5EA8	CBRC 026				RDH H AS,T+2	
289A	50A0	CBRC 027				RDH U AS,T	
		CBRC 028	*				
		CBRC 029	*				
		CBRC 030	*		BRANCH AND LINK	RX	
289C	2040	CBRC 031	RXOP45			SET S5	EXECUTION
289E	5C2B	CBRC 032	OP45A			VO=POH	RR
28A0	3245	CBRC 033				VO=VO\$K40	AND
28A2	E1B6	CBRC 034		044	EXECUT	BR IF S6=0	RX
28A4	D428	CBRC 035		037	OP45B	BR IF G01=0	BRANCH
28A6	224D	CBRC 036	OP45C			VO=VO+K40	AND
28A8	4D25	CBRC 037	OP45B			VO=PIXL\$VOH	LINK
28AA	2307	CBRC 038				V1=0	
28AC	55BB	CBRC 039				T1=G1H	
28AE	72A8	CBRC 040				STH V AS,T+2	
28B0	78A0	CBRC 041				STH I AS,T	
28B2	D187	CBRC 042		015	LEG 3	BR IF S5=1	
28B4	81B3	CBRC 043		CICY 005	ISTART N	N=S BITS67	
28B6	5892	CBRC 044	EXECUT			RDH I DA,AA	RESTORE EXECUTE INST. COUNTER
28B8	A8A6	CBRC 045		036	OP45C	BR	
		CBRC 046	*		EXECUTE OP CODE	44	
2C56	AB53	CBRC 047	RXOP44	CICY 045	ADDERR N	BR IF H1=NZ	
2C58	01E3	CBRC 048				Z=U1*-KOE	SPEC CHECK
2C5A	F0D4	CBRC 049		064	SPEC	BR IF LZNZ	
2C5C	E1D0	CBRC 050		062	OP44B	BR IF S6=0	EXECUTE CHECK
2C5E	7CC2	CBRC 051				STH P DA,BB	SAVE P
2C60	7892	CBRC 052				STH I DA,AA	
2C62	48C6	CBRC 053				I=U	
2C64	0020	CBRC 054				RST S6	
2C66	55BB	CBRC 055				T1=G1H	
2C68	C4CF	CBRC 056		065	RETRNC	BR IF Z=0	
2C6A	5498	CBRC 057				RDH G I+2	READ INSTRUCTION
2C6C	2B3B	CBRC 058				T1=T1+K03	ADJUST REG ADDRESS
2C6E	53A0	CBRC 059				RDB V1 AS,T+0	
2C70	6535	CBRC 060				G1=G1\$V1	
2C72	B461	CBRC 061		CICY 016	OPHI N	N=G0H	
2C50	2B33	CBRC 062	OP44B			T1=0\$K03	
2C52	9C72	CBRC 063		BPSW 004	PROGA	BR	
2C54	AE98	CBRC 064	SPEC	CICY 059	SPECHK	BR	
2C4E	A9A6	CBRC 065	RETRNC	CICY 008	ISTART 3	BR	
		CBRC 066	*		BRANCH ON COUNT	OP CODE 06	
2BCE	0040	CBRC 067	RROP06			RST S5	
2BD0	55B3	CBRC 068				T1=G1XH	
2BD2	C4DB	CBRC 069		074	OP4X	BR IF Z=0	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2BD4	5EA8	CBRC 070				RDH H AS,T+2	
2BD6	50A0	CBRC 071				RDH U AS,T+0	
		CBRC 072	*			BRANCH ON COUNT OP CODE 46	
2BD8	2040	CBRC 073	RXUP46			SET S5	
2BDCA	55BB	CBRC 074	OP4X			T1=G1H	
2BDC	56A8	CBRC 075				RDH D AS,T+2	
2BDE	52AA	CBRC 076				RDH V AS,T-2	
2BE0	000A	CBRC 077				RST S K=50	
2BE2	73A9	CBRC 078				V1C=V1-T0+C	DECREMENT
2BE4	72A9	CBRC 079				VOC=V0-T0+C	COUNT
2BE6	77A9	CBRC 080				D1C=D1-T0+C	BY
2BE8	76A9	CBRC 081				DOC=DO-T0+C	ONE
2BEA	76A8	CBRC 082				STH D AS,T+2	
2BEC	72AA	CBRC 083				STH V AS,T-2	
2BEE	D5F3	CBRC 084		086	CBRANC	BR IF S1=1	CHECK RESULT
2BF0	81B3	CBRC 085	NBRANC	CICY 005	ISTART N	N=S BITS67	
2BF2	D1FC	CBRC 086	CBRANC	085	NBRANC	BR IF S5=0	COUNT ONLY GO TO ICYC
2BF4	A886	CBRC 087		015	LEG 3	BR	
		CBRC 088	*			BRANCH INDEX HI,LO,EQUAL	THE FIRST WORD OF THIS ROUTINE IS ENTERED
		CBRC 089	*				BY OP CODES 86 AND 87
2CC8	8DC4	CBRC 090	RSOP86	CCOM 058	LSAVEB	BAL	SAVE I AND P
2CCA	55BB	CBRC 091				T1=G1H	
2CCC	58A8	CBRC 092				RDH I AS,T+2	
2CCE	52AA	CBRC 093				RDH V AS,T-2	INDEX IN IV
2CD0	55B3	CBRC 094				T1=G1XH	GET
2CD2	56A8	CBRC 095				RDH D AS,T+2	INCREMENT
2CD4	5CAA	CBRC 096				RDH P AS,T-2	FROM R3
2CD6	6D3B	CBRC 097				P1C=P1+V1	ADD
2CD8	6C2D	CBRC 098				POC=PO+VO+C	INDEX
2CDA	679D	CBRC 099				D1C=D1+I1+C	INTO
2CDC	668D	CBRC 100				DOC=DO+IO+C	INCREMENT
2CDE	3B15	CBRC 101				T1=T1\$K10	
2CE0	58A8	CBRC 102				RDH I AS,T+2	GET
2CE2	52AA	CBRC 103				RDH V AS,T-2	COMPARAND
2CE4	55BB	CBRC 104				T1=G1H	SET R1 ADDRESS
2CE6	76A8	CBRC 105				STH D AS,T+2	STORE
2CE8	7CAA	CBRC 106				STH P AS,T-2	SUM
2CEA	0004	CBRC 107				RST S2	
2CEC	30C2	CBRC 108				SET S K=90	
2CEE	7D3D	CBRC 109				P1C=P1%V1+C	COMPARE
2CF0	7C2D	CBRC 110				POC=PO%VO+C	COMPARAND
2CF2	779D	CBRC 111				D1C=D1%I1+C	TO THE
2CF4	768D	CBRC 112				DOC=DO%IO+C	SUM
2CF6	9258	CBRC 113		CCOM 066	LRSTRB	BAL	RESTORE I,P
2CF8	E4C1	CBRC 114		118	OVERFL	BR IF OVFL	BR IF OVERFLOW
2CFA	E5C3	CBRC 115	CHKANS	119	SIGN	BR IF S2=1	
2CFC	F047	CBRC 116	LHCHEK	121	GOBR	BR IF GO BIT7=1	EQUAL FIRST LO 2ND
2CFE	81B3	CBRC 117	NOBR	CICY 005	ISTART N	N=S BITS67	
2CC0	168D	CBRC 118	OVERFL			DO=DO#K80	OVERFLOW INVERT SIGN
2CC2	C67D	CBRC 119	SIGN	116	LHCHEK	BR IF DO BIT0=1	
2CC4	FG7F	CBRC 120		117	NOBR	BR IF GO BIT7=1	SUM IS HIGH
2CC6	A886	CBRC 121	GOBR	015	LEG 3	BR	

* CROSS REFERENCE FOR CSECT CBRC *

CBRC 005	CICY 129	
CBRC 011	CICY 170	
CBRC 012	CBRC 011	
CBRC 013	CBRC 006	CBRC 014
CBRC 015	CBRC 012	CBRC 017 CBRC 019 CBRC 042 CBRC 087 CBRC 121
CBRC 023	CICY 127	
CBRC 031	CICY 168	
CBRC 032	CBRC 025	
CBRC 036	CBRC 045	
CBRC 037	CBRC 035	
CBRC 044	CBRC 034	
CBRC 047	CICY 167	
CBRC 062	CBRC 050	
CBRC 064	CBRC 049	
CBRC 065	CBRC 056	
CBRC 067	CICY 128	
CBRC 073	CICY 169	
CBRC 074	CBRC 069	
CBRC 085	CBRC 086	
CBRC 086	CBRC 084	
CBRC 090	CICY 201	CICY 202
CBRC 116	CBRC 119	
CBRC 117	CBRC 120	
CBRC 118	CBRC 114	
CBRC 119	CBRC 115	
CBRC 121	CBRC 116	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CCOM 001	T			**** COMMON ROUTINES ****	
		CCOM 002	*				
		CCOM 003	*				
		CCOM 004	*				
		CCOM 005	*				
		CCOM 006	*	**		**THE FOLLING ROUTINES**	
		CCOM 007	*	**		**ARE OPERAND FETCHES**	
		CCOM 008	*				
		CCOM 009	*				
		CCOM 010	*				
		CCOM 011	*			** LINKED GET HALF ROUTINE **	
2DA8	AB53	CCOM 012	CXGETH	CICY 045	ADDERR N	BR IF H1=NZ	CHECK FOR VALID ADDRESS
2DAA	01E3	CCOM 013				Z=U1*-KOE	
2DAC	FOA6	CCOM 014		027	CXSPEC	BR IF LZNZ	CHECK BOUNDARY
2DAE	5E10	CCOM 015				RDH H U+0	
2DB0	CE23	CCOM 016		025	CXNEG	BR IF HO BIT0=1	CHECK SIGN
2DB2	2305	CCOM 017				V1=0\$K00	EXTEND PLUS SIGN
2DB4	5329	CCOM 018	CCSIGN			VO=V1	
2DB6	40E6	CCOM 019				U=H	
2DB8	55BB	CCOM 020	GETR1			T1=G1H	
2DBA	56A8	CCOM 021				RDH D AS,T+2	
2DBC	5EAA	CCOM 022				RDH H AS,T-2	
2DBE	100E	CCOM 023				RST S K=F0	RESET S BITS 0,1,2,3
2DC0	128E	CCOM 024				RTN	
2DA2	23F7	CCOM 025	CXNEG			V1=0\$KFF	
2DA4	C4B4	CCOM 026		018	CCSIGN	BR IF ZNZ	UNC BR ON ZNZ
2DA6	AE98	CCOM 027	CXSPEC	CICY 059	SPECHK	BR	
		CCOM 028	*				** OPERAND FETCH FOR LOADS **
037C	55B3	CCOM 029	LGET			T1=G1XH	
037E	56A8	CCOM 030				RDH D AS,T+2	
0380	5EAA	CCOM 031				RDH H AS,T-2	
0382	55BB	CCOM 032				T1=G1H	SET DESTINATION ADDR
0384	CCC8	CCOM 033				RST S1	RESET ZNZ STAT
0386	128E	CCOM 034				RTN	
		CCOM 035	*				** RR OPERAND FETCH **
0A70	55B3	CCOM 036	CRGET			T1=G1XH	
0A72	52A8	CCOM 037				RDH V AS,T+2	2ND OPERAND
0A74	50AA	CCOM 038				RDH U AS,T-2	
0A76	55BB	CCOM 039				T1=G1H	
0A78	56A8	CCOM 040				RDH D AS,T+2	1ST OPERAND
0A7A	5EAA	CCOM 041				RDH H AS,T-2	
0A7C	100E	CCOM 042				RST S K=F0	RESET S BITS 0,1,2,3
0A7E	128E	CCOM 043				RTN	
		CCOM 044	*				** RX OPERAND FETCH **
2770	5218	CCOM 045	CXGET			RDH V U+2	2ND OPERAND
2772	5010	CCOM 046				RDH U U+0	
2774	55BB	CCOM 047				T1=G1H \	
2776	56A8	CCOM 048				RDH D AS,T+2	1ST OPERAND
2778	5EAA	CCOM 049				RDH H AS,T-2	
277A	100E	CCOM 050				RST S K=F0	RESET S BITS 0,1,2,3
277C	128E	CCOM 051				RTN	
		CCOM 052	*				
		CCOM 053	*				

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CCOM 054	*				
		CCOM 055	*				
00C0	70A2	CCOM 056	LSAVE			STH U DA, AC	** LOCAL STORE SAVE **
00C2	74B2	CCOM 057	LSAVEA			STH G DA, AE	SAVE U
00C4	7CC2	CCOM 058	LSAVEB			STH P DA, BB	G
00C6	E1CC	CCOM 059		062	LSAVEC	BR IF S6=0	SAVE P
00C8	7892	CCOM 060				STH I DA, AA	TEST EXECUTE-BR IF YES
00CA	0020	CCOM 061				RST S6	I
00CC	128E	CCOM 062	LSAVEC			RTN	
		CCOM 063	*				** RESTORE LOCAL STORE **
1254	50A2	CCOM 064	LRSTR			RDH U DA, AC	RESTORE U
1256	54B2	CCOM 065	LRSTRA			RDH G DA, AE	G
1258	5CC2	CCOM 066	LRSTRB			RDH P DA, BB	P
125A	5892	CCOM 067	LRSTRC			RDH I DA, AA	I
125C	2020	CCOM 068				SET S6	
125E	128E	CCOM 069				RTN	
		CCOM 070	*				
307C	2B73	CCOM 071	FLGDEX			T1=0\$K07	FLAG DATA
307E	9C70	CCOM 072		BPSW 003	PROG	BR	EXECPTION
		CCOM 073	*				** LINKED COMPLIMENT ROUTINE **
		CCOM 074	*				
		CCOM 075	*				
0AE0	3002	CCOM 076	COMPL			SET S K=90	SET COMPLIMENT,CARRY
0AE2	7FFB	CCOM 077				HIC=0-H1+C	
0AE4	7EEB	CCOM 078				HOC=0-H0+C	
0AE6	777B	CCOM 079				DIC=0-D1+C	
0AE8	766B	CCOM 080				DOC=0-D0+C	
0AEA	100A	CCOM 081				RST S K=D0	RESET ZNZ,CARRY,COMP
0AEC	128E	CCOM 082				RTN	
		CCOM 083	*				** COMMON ADD,SUBTRACT,COMPARE ROUTINE **
		CCOM 084	*				
		CCOM 085	*				
0174	3002	CCOM 086	CSLOOP			SET S K=90	SET COMPLIMENT,CARRY
0176	7F1D	CCOM 087	CXLOOP			HIC=H1%U1+C	
0178	7E0D	CCOM 088				HOC=H0%U0+C	
017A	773D	CCOM 089				DIC=D1%V1+C	
017C	762D	CCOM 090				DOC=D0%V0+C	
017E	128E	CCOM 091				RTN	
		CCOM 092	*				*****8*****
		CCOM 093	*				
		CCOM 094	*				
		CCOM 095	*				*** CSW STORE ROUTINE *****
		CCOM 096	*				
		CCOM 097	*				
200E	26C7	CCOM 110	FULCSW			D0=0	
2010	577B	CCOM 114				D1=D1H	ADJUST UCW ADDR TO NAT KEY LOC
2012	3763	CCOM 115				D1=D1\$K06	D=UCW ADDR-00X6 ANY NATIVE
2014	5B60	CCOM 116				RDB T1 AS,D+0	GET NATIVE KEY T1=KKKK0000
2016	5B6B	CCOM 117				D0=T1H	D0=KKKK0000-NATIVE KEY-NOT COMM
2018	2707	CCOM 118	HELP			D1=0	ZERO OUT ADDRESS BYTE FOR CSW
201A	FOA5	CCOM 120		127	CSWADR	BR IF LZ=0	UNC BR
201C	4E52	CCOM 121	CLEAR			RDH H DC,9A	ENTRY FOR ALL ZERO CSW STORE
		CCOM 122	*				READ ZERO CONSTANT INTO H

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
201E	4252	CCOM 123	STATUS			RDH V DC,9A	** ENTRY FOR STAT STORE-ZERO REST OF CSW
		CCOM 124	*				READ ZERO CONSTANT INTO V
2020	4626	CCOM 125				D=V	
2022	4026	CCOM 126				U=V	
2024	2B45	CCOM 127	CSWADR			T1=0\$K40	
2026	42AF	CCOM 129				STPO=TO	DIXABLE CPU KEY
2028	76B8	CCOM 131				STH D T+2	KEY AND O,S
202A	72B8	CCOM 132				STH V T+2	NEXT CCW ADDR
202C	6AA4	CCOM 133				T=T+2	
202E	70B0	CCOM 134				STH U T+0	RES COUNT
2030	2B47	CCOM 135	STATUS			T1=0\$K44	ENTRY FOR STATUS ONLY
2032	42AF	CCOM 137				STPO=TO	DIXABLE CPU KEY
2034	7EB0	CCOM 139				STH H T+0	STATUS=UNIT,CHAN.
2036	5282	CCOM 141				RDH V DA,A8	GET CPU KEY
2038	423F	CCOM 142				STPO=V1	PUT CPU KEY BACK IN Q
203A	C5BE	CCOM 144		146	NORTN	BR IF SO=0	SO=1 FOR RTN FROM THIS ROUTINE
203C	128E	CCOM 145				RTN	
203E	ADF4	CCOM 146	NORTN	157	CC1B	BR	GO RST MODE ,SET CC1
		CCOM 147	*				
		CCOM 148	*				
		CCOM 149	*				
		CCOM 150	*				
		CCOM 151	*				
		CCOM 152	*				****COMMON CONDITION CODE SET****
		CCOM 153	*				
3362	2400	CCOM 154	CCOB			SET MODE K=00	
3364	A8DC	CCOM 155		CICY 003	CHECKX	BR	
		CCOM 156	*				
2DF4	2400	CCOM 157	CC1B			SET MODE K=00	
2DF6	2C55	CCOM 158	CC1C			PO=0\$K50	SET CC1-BITS2,3=COND CODE
2DF8	81B3	CCOM 159		CICY 005	ISTART N	N=S BITS67	
		CCOM 160	*				
2C74	2400	CCOM 161	CC2B			SET MODE K=00	
2C76	2C25	CCOM 162	CC2C			PO=0\$K20	SET CC2
2C78	81B3	CCOM 163		CICY 005	ISTART N	N=S BITS67	
		CCOM 164	*				
2AF4	2400	CCOM 165	CC3B			SET MODE K=00	
2AF6	2C75	CCOM 166	CC3C			PO=0\$K70	SET CC3
2AF8	81B3	CCOM 167		CICY 005	ISTART N	N=S BITS67	
		CCOM 168	*				
		CCOM 169	*				*****
		CCOM 170	*				
		CCOM 171	*				** STORE H1 ROUTINE **
		CCOM 172	*				
32F8	2BB1	CCOM 173	STORH1			T1=0-KOB	
32FA	7FA0	CCOM 174				STB H1 AS,T+0	
32FC	128E	CCOM 175				RTN	
		CCOM 176	*				
		CCOM 177	*				*****
		CCOM 178	*				
		CCOM 179	*				** RESTORE CPU KEY ROUTINE **
		CCOM 180	*				
27FA	5682	CCOM 181	RSTRKY			RDH D DA,A8	GET CPU KEY

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
27FC	427F	CCOM 182				STPO=D1	RESTORE CPU KEY INTO Q
27FE	128E	CCOM 183				RTN	
		CCOM 184	*			*****	
		CCOM 185	*			*** RESTORE HI ROUTINE ***	
3354	2BB1	CCOM 186	RESTRH			T1=0-KOB	T=00F4
3356	5FA0	CCOM 187				RDB HI AS,T+0	
3358	128E	CCOM 188				RTN	

						* CROSS REFERENCE FOR CSECT CCOM *	

CCOM 012	CBIN 022	CBIN 038	CBIN 051	CMLT 014			
CCOM 018	CCOM 026						
CCOM 020	CNVR 013						
CCOM 025	CCOM 016						
CCOM 027	CCOM 014						
CCOM 029	CDVD 004	CLST 005	CLST 018	CLST 030	CLST 040		
CCOM 036	CBIN 005	CBIN 026	CBIN 042	CBIN 055	CBIN 076	CLOG 036	CLOG 069 CLOG 110 CLOG 140 CMLT 011
CCOM 045	CBIN 008	CBIN 030	CBIN 046	CBIN 059	CBIN 080	CLOG 039	CLOG 073 CLOG 114 CLOG 144 CMLT 004
CCOM 056	CDMD 009	ERDR 067					
CCOM 057	CDVD 011	CFCY 016	CFCY 036	CMLT 015	CNVR 009		
CCOM 058	CBRC 090	DPTS 045	DYPE 004	DYPE 113	DYPE 226	ERRQ 021	ERRQ 066
CCOM 062	CCOM 059						
CCOM 064	CDMD 054	CDMD 059	CDMD 139				
CCOM 065	CDVD 086	CDVD 098	CMLT 070	CNVR 111			
CCOM 066	BSWI 072	CBRC 113	CFAD 072	CNVR 047	DPTR 070	DPTS 078	DYPE 154 ERDR 070
CCOM 067	CFAD 127	CFCY 068	CFHA 033	CFLS 042			
CCOM 071	CDMD 055	CNVR 073	CSAS 025	CTRT 085			
CCOM 076	CDVD 018	CLST 007	CLST 032	CLST 042	CNVR 017		
CCOM 086	CBIN 009	CBIN 047	CBIN 081	CLOG 040			
CCOM 087	CBIN 031	CBIN 060					
CCOM 110	DCHN 072	DPTT 041					
CCOM 123	DCHN 090	DCLC 057	DPTT 015	DPTT 054	FILE 118	FINT 041	
CCOM 127	CCOM 120	DCLT 040	FILE 018	FINT 009			
CCOM 135	DCLA 227	DCLC 168	DCLR 011	DPTS 079	DPTT 068	FILE 128	FILX 437
CCOM 146	CCOM 144						
CCOM 154	DCLA 041	DCLA 228	DCLC 065	DCLH 021	DPTT 011	DYPE 202	ERRQ 011 FILE 030 FILE 094 FILX 189 FILX 328 FILX 445
CCOM 157	CCOM 146	DCLA 042					
CCOM 161	DCLA 060	DCLA 285	DCLB 005	DCLT 008	DPTT 025	FILE 026	
CCOM 165	DCLC 037	DYPE 215					
CCOM 173	BSTP 007	BSWI 009					
CCOM 181	BMCK 074	BSWI 043	BSWI 069	CSTS 053			
CCOM 186	BSWI 075	DCLT 041					

COMD -- DECIMAL MULTIPLY-DIVIDE ROUTINES

ENTRY POINT

SSOPFC - THIS IS THE COMMON ENTRY POINT FOR BOTH MULTIPLY
DECIMAL AND DIVIDE DECIMAL.

 . MULT DECIMAL . FC . L1 L2 . B1 D1 . D1 D1 . B2 D2 . D2 D2 .
 . DIV DECIMAL . FD . L1 L2 . B1 D1 . D1 D1 . B2 D2 . D2 D2 .

OBJECTIVES

MULTIPLY - THE PRODUCT OF THE MULTIPLIER AND THE
MULTPLICAND REPLACES THE MULTIPLICAND IN THE
1ST OPERAND LOCATION.

THE MULTIPLIER SIZE IS LIMITED TO 15 DIGITS
AND SIGN -- LENGTH CODE OF 7 IN L2. ALSO, L2
MUST NOT BE LARGER THAN OR EQUAL TO L1.

THE MULTIPLICAND MUST HAVE HIGH-ORDER ZERO
BYTES EQUAL TO OR GREATER THAN THE NUMBER OF
BYTES IN THE MULTIPLIER FIELD.

THE MAXIMUM PRODUCT SIZE IS 31 DIGITS. AT
LEAST ONE HIGH-ORDER DIGIT OF THE PRODUCT
FIELD IS ZERO.

DIVIDE - THE DIVIDEND IS DIVIDED BY THE DIVISOR. THE
QUOTIENT AND THE REMAINDER REPLACE THE DIVIDEND
IN THE 1ST OPERAND LOCATION.

THE QUOTIENT FIELD IS PLACED IN THE LEFT
PORTION OF THE DIVIDEND FIELD. THE REMAINDER
IS THE SAME SIZE AS THE DIVISOR AND OCCUPIES
THE LOW-ORDER BYTES OF THE DIVIDEND FIELD.

THE DIVISOR SIZE IS LIMITED TO 15 DIGITS AND
SIGN -- LENGTH CODE OF 7 IN L2. ALSO, L2 MUST
NOT BE LARGER THAN OR EQUAL TO L1.

DESCRIPTION

THE ROUTINE CHECKS OPERAND SIZE AND SETS THE SIGN VALUE.
THEN, TWO COPIES OF THE MULTIPLIER OR DIVISOR ARE STORED.

	AUX STORAGE FIELD	POINTER FOR LOW-ORDER BYTE
STRAIGHT COPY	0018-001F	00BC
SKewed COPY	0038-00BF	00BE

DESCRIPTION CONTINUED

MULTIPLY OP

1. TEST FOR ENOUGH ZEROES IN MULTIPLICAND FIELD.
2. SHIFT THE MOST SIGNIFICANT BYTE AND THE FOLLOWING
BYTES OF THE MULTIPLICAND TO THE HIGH-ORDER POSITIONS
OF THE MULTIPLICAND FIELD. SUPPLY ZEROES FOR THE
VACATED BYTES, REMOVE SIGN.
3. STARTING WITH THE LOW-ORDER, TEST EACH DIGIT OF
MULTIPLICAND. TRUE ADD IF UNDER 5, SUBTRACT IF OVER
4. ENTER CALCULATE LOOP FOR EACH DIGIT. FOR THE 1ST
DIGIT, THE STRAIGHT MULTIPLIER WITH 90 AS THE HIGH-
ORDER BYTE IS T/C ADDED TO THE PRODUCT FIELD. THE 9
IS IN ALIGNMENT WITH THE TESTED MULTIPLICAND DIGIT.
T/C ADD CYCLES ARE CONTINUED UNTIL TEST DIGIT IS 0
FOR ADD OR 9 FOR SUBTRACT.
ALTERNATE MULTIPLICAND DIGITS USE THE SKEWED MULTI-
PLIER WITH 09 AS THE HIGH-ORDER BYTE UNLESS THE TEST
DIGIT GOES TO 0 FOR ADD OR 9 FOR SUBTRACT. FOR THIS
CASE, THE MULTIPLIER IS NOT SWITCHED.
4. DO SPECIAL SKEWED ADD IF NECESSARY, INSERT SIGN, AND
END THE OPERATION.

DIVIDE OP

1. DO A TRIAL SUBTRACT TO DETERMINE IF THE QUOTIENT AND
REMAINDER CAN BE CONTAINED IN THE DIVIDEND FIELD.
SKEWED DIVISOR IS USED. A CARRY OUT OF THE HIGH-ORDER
POSITION INDICATES A DIVIDE CHECK. IF DIVIDE CHECK
OCCURS, SKEWED DIVISOR IS ADDED BACK TO RESTORE THE
DIVIDEND AND ROUTINE EXITS TO TAKE PROGRAM INTERRUPT.
2. FOR 1ST CALCULATE CYCLE, STRAIGHT DIVISOR WITH 90 AS
THE HIGH-ORDER BYTE IS ADDED TO THE DIVIDEND. THE 90
IS IN ALIGNMENT WITH THE HIGH-ORDER DIVIDEND BYTE.
THE ADD CYCLES CONTINUE UNTIL DIGIT TO THE RIGHT OF
THE 9 GOES TO ZERO.
3. SKEWED DIVISOR WITH 09 AS THE HIGH-ORDER BYTE IS
SUBTRACTED FROM THE PREVIOUS RESULT WITH ALIGNMENT
SHIFTED ONE DIGIT. SUBTRACT CYCLES CONTINUE UNTIL
DIGIT TO THE RIGHT OF THE 9 GOES TO NINE
4. ALTERNATE STRAIGHT AND SKEWED CYCLES ARE TAKEN WITH
DIVISOR SHIFTING ONE DIGIT FOR EACH CHANGE.
OPERATION ENDS WHEN THE CYCLE THAT INCLUDES THE
LEFTMOST POSITION OF THE DIVIDEND IS COMPLETE.
5. QUOTIENT SIGN IS INSERTED INTO LOW-ORDER 4 BITS OF
QUOTIENT. SIGN OF REMAINDER IS INSERTED INTO LOW-
ORDER 4 BITS OF REMAINDER.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CDMD 001	T			DECIMAL MULTIPLY DIVIDE ROUTINES	
2582	E187	CDMD 002	SSOPFC	004	NOFILC	BR IF S6=1	
2584	1445	CDMD 003				GO=GO*-K40	FIX ILC FOR EXECUTE
2586	C10A	CDMD 004	NOFILC	006	LENCHK	BR IF G14=0	LENGTH CHECK OVER 7
2588	AE98	CDMD 005	FLGERR	CICY 059	SPECHK	BR	
258A	5575	CDMD 006	LENCHK			D1=G1XL	CHECK IF
258C	77E3	CDMD 007				D1=D1-H0	L2 LARGER
258E	EC88	CDMD 008		005	FLGERR	BR IF HZNZ	THAN L1
2590	8DC0	CDMD 009		CCCM 056	LSAVE	BAL	
2592	573C	CDMD 010				RDB D1 V	READ 2ND OPRAND
2594	A4EC	CDMD 011		203	SINSET	BAL	GO SET THE SIGN VALUE
2596	5710	CDMD 012				RDB D1 U	READ 1ST OPRAND
2598	3415	CDMD 013				GO=GO\$K10	START WITH REMAINDER SIGN +
259A	A4EC	CDMD 014		203	SINSET	BAL	GO SET THE SIGN VALUE
259C	10CE	CDMD 015				RST S K=FC	RESET S REG 0,1,2,3,4,5
259E	558D	CDMD 016				T1=GIL	-BUILD THE
25A0	3B83	CDMD 017				T1=T1\$K08	STRAIGHT AND SKEWED ADDRESSES
25A2	3B15	CDMD 018				T1=T1\$K10	AND PUT THEM IN AUX
25A4	7AE2	CDMD 019				STH T DA,BC	OOBC-STRAIGHT
25A6	4CA6	CDMD 020				P=T	OOEE-SKEWED
25A8	3B25	CDMD 021				T1=T1\$K20	
25AA	7AF2	CDMD 022				STH T DA,BE	
25AC	5F3A	CDMD 023				RDB H1 V-1	-THIS STORES THE DIVISOR
25AE	5FFB	CDMD 024				H1=H1H	PLIER IN AUX STRAIGHT AND
25B0	7FAF	CDMD 025				H1C=H1@TO+C	SKEWED
25B2	7FCA	CDMD 026				STB H1 AS,P-1	
25B4	5FF1	CDMD 027				H1=H1X	
25B6	C946	CDMD 028		036	SSKEW	BR IF P14=0	TAKE THIS BR IF 1 POS. FIELD
25B8	573A	CDMD 029	SMORE			RDB D1 V-1	
25BA	47F3	CDMD 030				H1=D1XH+H1L	
25BC	77AF	CDMD 031				D1C=D1@TO+C	
25BE	7FAA	CDMD 032				STB H1 AS,T-1	
25C0	77CA	CDMD 033				STB D1 AS,P-1	
25C2	57F5	CDMD 034				H1=D1XL	
25C4	C939	CDMD 035		029	SMORE	BR IF P14=1	
25C6	7FAA	CDMD 036	SSKEW			STB H1 AS,T-1	STORE LAST SKEWED CHAR.
25C8	55E5	CDMD 037				H0=G1XL	-THIS ADJUSTS THE ADDRESS 1ST OP
25CA	55FD	CDMD 038				H1=GIL	TO THE LEFT MOST POS.
25CC	7EF1	CDMD 039				H0=H0-H1+1	
25CE	2002	CDMD 040				SET S3	
25D0	71E9	CDMD 041				UIC=UI-H0+C	
25D2	70A9	CDMD 042				UOC=UO-TO+C	
25D4	4206	CDMD 043				V=U	
		CDMD 044	*				
25D6	F001	CDMD 045		047	SOUT	BR IF G07=1	BREAK OUT
25D8	8B82	CDMD 046		106	MULTY	BR	THE OP
2580	A51C	CDMD 047	SOUT	052	DIVIDE	BR	CODES
		CDMD 048	*				
		CDMD 049	*			AT THIS POINT THE DIVIDE OPERATION	
		CDMD 050	*			IS STARTED IT GOES BACK TO A COMMON	
		CDMD 051	*			POINT WITH MULTIPLY TO SET THE RESULT SIGN	
251C	9AEE	CDMD 052	DIVIDE	213	DVLOOP	BAL	GO SEE IF ANS. WILL FIT
251E	C1A4	CDMD 053		056	VALIDA	BR IF S4=0	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2520	9254	CDMD 054	DATEXP	CCOM 064	LRSTR	BAL	THIS RESTORES LOCAL STORAGE AND FLAGS A DATA EXCEPTION INTERUPT IF S3=1 A DIVIDE EXCEPTION EXIST ADD BACK IN THE VALUE SUB AND FLAG DIV. EXCEP. INTER.
2522	807C	CDMD 055		CCOM 071	FLGDEX	BR	
2524	F5B0	CDMD 056	VALIDA	062	VALIFL	BR IF S3=0	
2526	1002	CDMD 057				RST S K=90	
2528	9AF0	CDMD 058		214	DVLOPB	BAL	
252A	9254	CDMD 059		CCOM 064	LRSTR	BAL	
252C	28B3	CDMD 060				T1=0\$K0B	
252E	9C70	CDMD 061		BPSW 003	PROG	BR	
2530	56A2	CDMD 062	VALIFL			RDH D DA, AC	
2532	5D70	CDMD 063				RDB P1 D	
2534	5DDB	CDMD 064				P1=P1H	
2536	7D70	CDMD 065				STB P1 D	
2538	2993	CDMD 066				I1=0\$K09	
253A	5981	CDMD 067				I0=I1X	
253C	5004	CDMD 068	ENTYA			U=U+1	
253E	1042	CDMD 069	ENTYB			RST S K=94	
2540	56E2	CDMD 070				RDH D DA, BC	
2542	9AF2	CDMD 071		215	DVLOPC	BAL	
2544	5D30	CDMD 072				RDB P1 V	
2546	7D8F	CDMD 073				P1C=P1@I0+C	
2548	F081	CDMD 074		077	OTHLOP	BR IF LZ=0	
254A	7D30	CDMD 075				STB P1 V	
254C	A53E	CDMD 076		069	ENTYB	BR	
2500	7D30	CDMD 077	OTHLOP			STB P1 V	
2502	7EA3	CDMD 078				H0=H0-T0	
2504	C4CF	CDMD 079		091	CHECK	BR IF Z=0	
2506	9AEE	CDMD 080	OTALOP	213	DVLOOP	BAL	
2508	0D9D	CDMD 081				Z=P1@K90	
250A	EC8E	CDMD 082		084	ARDN	BR IF HZNZ	
250C	2040	CDMD 083				SET S5	
250E	5D30	CDMD 084	ARDN			RDB P1 V	
2510	5D7B	CDMD 085				D1=P1H	
2512	7D5F	CDMD 086				P1C=P1@I1+C	
2514	47DB	CDMD 087				P1=D1H+P1L	
2516	7D30	CDMD 088				STB P1 V	
2518	D186	CDMD 089		080	OTALOP	BR IF S5=0	
251A	A53C	CDMD 090		068	ENTYA	BR	
254E	C1A1	CDMD 091	CHECK	054	DATEXP	BR IF S4=1	
2550	5CC2	CDMD 092				RDH P DA, B8	
2552	2FC3	CDMD 093				H1=0\$K0C	
2554	C958	CDMD 094		096	ARDNB	BR IF P14=0	
2556	1F6B	CDMD 095				H1=H1@K06	
2558	E45D	CDMD 096	ARDNB	098	ARDNC	BR IF G02=1	
255A	3F13	CDMD 097				H1=H1\$K01	
255C	5730	CDMD 098	ARDNC			RDB D1 V	
255E	67F5	CDMD 099				D1=D1\$H1	
2560	7730	CDMD 100				STB D1 V	
2562	6443	CDMD 101				GO=GO+GO	
2564	8BC2	CDMD 102		138	MEND	BR	
		CDMD 103	*				
		CDMD 104	*				
		CDMD 105	*				
0B82	571A	CDMD 106	MULTPY			RDB D1 U-1	CHECK

THIS
SETS
THE
QUO
SIGN

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0B84	A317	CDMD 107		150	EREXCP N	BR IF D1=NZ	FIELD FOR
0B86	2FFF	CDMD 108				H1=H1+KFF	ENOUGH ZEROS
0B88	F483	CDMD 109		106	MULTPY	BR IF AC=1	
0B8A	5004	CDMD 110				U=U+1	
0B8C	5224	CDMD 111				V=V+1	
0B8E	5EF9	CDMD 112				H1=H0	
0B90	2B07	CDMD 113				T1=0	
0B92	E597	CDMD 114		116	MNZERO	BR IF S2=1	
0B94	2040	CDMD 115				SET S5	
0B96	0004	CDMD 116	MNZERO			RST S2	
0B98	77AF	CDMD 117	MLUPA			D1C=D1aT0+C	STORE THE
0B9A	5730	CDMD 118				R0B D1 V	1ST
0B9C	D1A1	CDMD 119		121	MZER	BR IF S5=1	OP
0B9E	7718	CDMD 120				STB D1 U+1	SHIFTED
0BA0	7B38	CDMD 121	MZER			STB T1 V+1	TO
0BA2	2FFF	CDMD 122				H1=H1+KFF	LEFT
0BA4	C498	CDMD 123		117	MLUPA	BR IF ZNZ	
0BA6	5006	CDMD 124				U=U-1	
0BA8	5226	CDMD 125				V=V-1	
0BAA	D1C3	CDMD 126		138	MEND	BR IF S5=1	
0BAC	577B	CDMD 127				D1=D1H	RID OG
0BAE	7710	CDMD 128				STB D1 U	SIGN
0BB0	77AF	CDMD 129				D1C=D1aT0+C	
0BB2	C181	CDMD 130		150	EREXCP 0	BR IF S4=1	
0BB4	E5C2	CDMD 131		138	MEND	BR IF S2=0	
0BB6	B1BC	CDMD 132		151	MSTART	BR	
0BB8	0B9D	CDMD 133	ENDING			Z=T1aK90	
0BBA	EOC2	CDMD 134		138	MEND	BR IF HZNZ	
0BBC	4086	CDMD 135				U=I	
0BBE	1002	CDMD 136				RST S K=90	
0BC0	9AF0	CDMD 137		214	DVLOPB	BAL	
0BC2	54F9	CDMD 138	MEND			H1=GO	SET THE
0BC4	9254	CDMD 139		CCOM 064	LRSTR	BAL	SIGN
0BC6	27C3	CDMD 140				D1=0aK0C	FOR
0BC8	C94C	CDMD 141		143	MASCI	BR IF P14=0	REMAIN
0BCA	176B	CDMD 142				D1=D1aK06	AND
0BCC	EF51	CDMD 143	MASCI	145	MPLUS	BR IF H12=1	PRODUCT
0BCE	3713	CDMD 144				D1=D1aK01	
0BD0	5F10	CDMD 145	MPLUS			RDB H1 U	
0BD2	6F75	CDMD 146				H1=H1aD1	
0BD4	7F10	CDMD 147				STB H1 U	
0BD6	2A07	CDMD 148				T0=0	
0BD8	A8DE	CDMD 149		CICY 004	CHECK	BR	
0B80	A520	CDMD 150	EREXCP 0	054	DATEXP	BR	
31BC	2040	CDMD 151	MSTART			SET S5	
31BE	4826	CDMD 152				I=V	THIS
31C0	57B9	CDMD 153	MSB			T1=D1	DOES
31C2	C5C6	CDMD 154	MSJKA	156	MSA	BR IF S0=0	THE
31C4	0B9F	CDMD 155				Z=T1aK99	ACTUAL
31C6	0C8E	CDMD 156	MSA			RST S K=78	MULTIPLY
31C8	D1DA	CDMD 157		174	MSLOW	BR IF S5=0	SO 1-SUB
31CA	EOEE	CDMD 158	MSHI	166	MSC	BR IF HZNZ	0-ADD
31CC	7B1A	CDMD 159	MSSTOR			STB T1 U-1	S5 0-LOW HALF

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
31CE	5886	CDMD 160				I=I-1	1-HI HALF
31D0	5710	CDMD 161				RDB D1 U	
31D2	0040	CDMD 162				RST S5	HO=NUMBER OF BYTES
31D4	2EFF	CDMD 163				HO=HO+KFF	
31D6	C4C0	CDMD 164		153	MSB	BR IF ZNZ	
31D8	8BB8	CDMD 165		133	ENDING	BR	
31EE	2040	CDMD 166	MSC			SET S5	
31F0	2A95	CDMD 167				T0=0\$K90	
31F2	56E2	CDMD 168				RDH D DA, BC	
31F4	57C9	CDMD 169				PO=D1	
31F6	100C	CDMD 170				RST S0	
31F8	CBB9	CDMD 171				Z=T1+K80	
31FA	F4EB	CDMD 172		182	SUBCYC	BR IF AC=1	
31FC	A74C	CDMD 173		185	CALLOP	BR	
31DA	FOCB	CDMD 174	MSLOW	158	MSHI	BR IF LZ=0	
31DC	58BD	CDMD 175				T1=T1L	
31DE	2A93	CDMD 176				T0=0\$K09	
31E0	56F2	CDMD 177				RDH D DA, BE	
31E2	57C9	CDMD 178				PO=D1	
31E4	1000	CDMD 179				RST S0	
31E6	CBB1	CDMD 180				Z=T1+K0B	
31E8	EOED	CDMD 181		183	ADDCYC	BR IF HZ=0	
31EA	3002	CDMD 182	SUBCYC			SET S K=90	
31EC	A74C	CDMD 183	ADDCYC	185	CALLOP	BR	
274A	5C79	CDMD 184	MSJK			D1=PO	
274C	4286	CDMD 185	CALLOP			V=I	
274E	5D30	CDMD 186	MSCLOP			RDB P1 V	CALCULATE LOOP
2750	5F6A	CDMD 187				RDB H1 AS, D-1	
2752	7DFF	CDMD 188				P1C=P1@H1+C	
2754	7D3A	CDMD 189				STB P1 V-1	
2756	C34F	CDMD 190		186	MSCLOP	BR IF D14=1	
2758	7BAF	CDMD 191				T1C=T1@T0+C	
275A	0002	CDMD 192				RST S3	
275C	C5E2	CDMD 193		196	MSJA	BR IF S0=0	
275E	0B9F	CDMD 194				Z=T1@K99	
2760	2002	CDMD 195				SET S3	
2762	D1E8	CDMD 196	MSJA	199	MSKY	BR IF S5=0	
2764	ECCA	CDMD 197		184	MSJK	BR IF HZNZ	
2766	B1CC	CDMD 198		159	MSSTOR	BR	
2768	FOCA	CDMD 199	MSKY	184	MSJK	BR IF LZNZ	
276A	5710	CDMD 200				RDB D1 U	
276C	478B	CDMD 201				T1=D1H+T1L	
276E	B1C2	CDMD 202		154	MSJKA	BR	
24EC	5773	CDMD 203	SINSET			D1=D1XH	SIGN CHECK AND SET ROU
24EE	0769	CDMD 204				Z=D1+K60	GO IS USED TO PUT THE SIGNS
24F0	F4F5	CDMD 205		207	SIGVAL	BR IF AC=1	
24F2	A520	CDMD 206		054	DATEXP	BR	BRANCH TO DATA EXCEPTION
24F4	F77C	CDMD 207	SIGVAL	211	EXIT	BR IF D13=0	1010- PLUS 1101-MINUS
24F6	0719	CDMD 208				Z=D1+K10	1011- MINUS 1110-PLUS
24F8	F4FD	CDMD 209		211	EXIT	BR IF AC=1	1100- PLUS 1111-PLUS
24FA	143D	CDMD 210				GO=GO@K30	GO BIT 2 0=ANS MINUS 1-PLS
24FC	128E	CDMD 211	EXIT			RTN	GO BOT 3 0=REM NEG 1-PLUS
		CDMD 212	*				

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1AEE	3002	CDMD 213	DVLOOP			SET S K=90	SET S 0,3 FOR SUB CYCLE
1AF0	56F2	CDMD 214	DVLOPB			RDH D DA,BE	GET THE SKEWED ADDRESS
1AF2	4206	CDMD 215	DVLOPC			V=U	SET UP V FOR LOOP
1AF4	5D30	CDMD 216	ALOOP			RDB P1 V	ACTUAL LOOP
1AF6	5F6A	CDMD 217				RDB H1 AS,D-1	FOR DOING
1AF8	7DFF	CDMD 218				PIC=P1@H1+C	THE ADD OR
1AFA	7D3A	CDMD 219				STB P1 V-1	SUB FUNCTION
1AFC	C375	CDMD 220		216	ALOOP	BR IF D14=1	
1AFE	128E	CDMD 221				RTN	

 * CROSS REFERENCE FOR CSECT CDMD *

CDMD 002	CICY 253	CICY 254	
CDMD 004	CDMD 002		
CDMD 005	CDMD 008		
CDMD 006	CDMD 004		
CDMD 029	CDMD 035		
CDMD 036	CDMD 028		
CDMD 047	CDMD 045		
CDMD 052	CDMD 047		
CDMD 054	CDMD 091	CDMD 150	CDMD 206
CDMD 056	CDMD 053		
CDMD 062	CDMD 056		
CDMD 068	CDMD 090		
CDMD 069	CDMD 076		
CDMD 077	CDMD 074		
CDMD 080	CDMD 089		
CDMD 084	CDMD 082		
CDMD 091	CDMD 079		
CDMD 096	CDMD 094		
CDMD 098	CDMD 096		
CDMD 106	CDMD 046	CDMD 109	
CDMD 116	CDMD 114		
CDMD 117	CDMD 123		
CDMD 121	CDMD 119		
CDMD 133	CDMD 165		
CDMD 138	CDMD 102	CDMD 126	CDMD 131 CDMD 134
CDMD 143	CDMD 141		
CDMD 145	CDMD 143		
CDMD 150	CDMD 107	CDMD 130	
CDMD 151	CDMD 132		
CDMD 153	CDMD 164		
CDMD 154	CDMD 202		
CDMD 156	CDMD 154		
CDMD 158	CDMD 174		
CDMD 159	CDMD 198		
CDMD 166	CDMD 158		
CDMD 174	CDMD 157		
CDMD 182	CDMD 172		
CDMD 183	CDMD 181		
CDMD 184	CDMD 197	CDMD 199	
CDMD 185	CDMD 173	CDMD 183	
CDMD 186	CDMD 190		

* CROSS REFERENCE FOR CSECT CDMD *

CDMD 196	CDMD 193		
CDMD 199	CDMD 196		
CDMD 203	CDMD 011	CDMD 014	CNVR 070
CDMD 207	CDMD 205		
CDMD 211	CDMD 207	CDMD 209	
CDMD 213	CDMD 052	CDMD 080	
CDMD 214	CDMD 058	CDMD 137	
CDMD 215	CDMD 071		
CDMD 216	CDMD 220		

CDVD DESCRIPTIVE TEXT

```
*****
*          *          *          *          *          *
* ENTRY  * OPERATION * BYTE 1 * BYTE 2 * BYTE 3 * BYTE 4 *
*-----*-----*-----*-----*-----*-----*
* RROP1D * BINARY DIVIDE * 1D * R1 R2 * * *
* RKOP5D * BINARY DIVIDE * 5D * R1 X2 * B2 D2 * D2 *
*****
```

EXECUTION

```
READOUT THE DIVISOR
CHECK FOR EVEN DIVIDEND ADDRESS
  IF ODD ADDRESS, FLAG SPECIFICATION EXCEPTION
  IN CICY ROUTINE AND GO STORE INTERRUPT CODE
  IN PSW.
READOUT THE DIVIDEND
  IF DIVIDEND MINUS, COMPLEMENT IT.
  IF DIVISOR MINUS, COMPLEMENT IT.
DO A TRIAL SUBTRACT
  IF QUOTIENT WILL FIT, DO THE DIVIDE.
  IF QUOTIENT TOO LARGE, FLAG A DIVIDE EXCEPTION
  AND GO TO STORE INTERRUPT CODE IN PSW.
DO THE SUBTRACT
  THE QUOTIENT AND REMAINDER ARE PLACED IN THE
  FIRST OPERAND LOCATION.
```

OBJECTIVES

THE DIVIDEND (1ST OPERAND) IS DIVIDED BY THE DIVISOR (2ND OPERAND). THE QUOTIENT AND REMAINDER ARE PLACED IN THE FIRST OPERAND LOCATION.

THE DIVIDEND IS A 64 BIT SIGNED INTEGER THAT MUST BE LOCATED IN AN EVEN-ODD PAIR OF GENERAL PURPOSE REGISTERS. THE DIVISOR IS A 32 BIT SIGNED INTEGER. THE QUOTIENT SIGN IS DETERMINED BY THE RULES OF ALGEBRA.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CDVD 001	T			BINARY DIVIDE ROUTINES	
19F8	E1FD	CDVD 002	KROP1D	004	NEQLAB	BR IF S6=1	
19FA	3485	CDVD 003				GO=GO\$K80	
19FC	837C	CDVD 004	NEQLAB	CCOM 029	LGET	BAL	
19FE	A21E	CDVD 005		009	MEET	BR	
2218	5618	CDVD 006	KXOP5D			RDH D U+2	READ
221A	5E10	CDVD 007				RDH H U	OUT DIV
221C	558B	CDVD 008				T1=G1H	
221E	F522	CDVD 009	MEET	011	MEETA	BR IF G13=0	
2220	AE58	CDVD 010		CICY 059	SPECHK	BR	
2222	80C2	CDVD 011	MEETA	CCOM 057	LSAVEA	BAL	
2224	54A8	CDVD 012				RDH G AS,T+2	READ
2226	58AA	CDVD 013				RDH I AS,T-2	OUT
2228	3815	CDVD 014				T1=T1\$K10	THE
222A	52A8	CDVD 015				RDH V AS,T+2	DVD
222C	50AA	CDVD 016				RDH U AS,T-2	
222E	C634	CDVD 017		020	MLETEB	BR IF D00=0	
2230	8AE0	CDVD 018		CCOM 076	COMPL	BAL	MINUS
2232	3813	CDVD 019				T1=T1\$K01	DIV
2234	C43A	CDVD 020	MEETB	023	MEETD	BR IF G00=0	
2236	9708	CDVD 021		088	COMDVD	BAL	MINUS
2238	3B23	CDVD 022				T1=T1\$K02	DVD
223A	4C86	CDVD 023	MEETD			P=I	SETUP
223C	30C2	CDVD 024				SET S K=90	AND DO

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
223E	7DFD	CDVD 025				PLC=P1%H1+C	TRIAL
2240	7CED	CDVD 026				POC=P0%H0+C	SUBTRACT
2242	4C46	CDVD 027				P=G	
2244	7D7D	CDVD 028				PLC=P1%D1+C	
2246	7C6D	CDVD 029				POC=P0%D0+C	
2248	F5CC	CDVD 030		032	MEETC	BR IF S3=0	BRANCJ OK
224A	AEFC	CDVD 031		098	DIVCHK	BR	
224C	4C52	CDVD 032	MEETC			KDH P DC,9A	
224E	10CE	CDVD 033				RST S K=FC	
2250	A6F8	CDVD 034		047	SHIFTA	BAL	
2252	825A	CDVD 035		037	ARND	BR	
0258	A6F6	CDVD 036	INSERT	046	SHIFT	BAL	
025A	3002	CDVD 037	ARND			SET S K=90	
025C	79FD	CDVD 038	COMPUT			IIC=I1%H1+C	
025E	78ED	CDVD 039				IOC=I0%H0+C	
0260	757D	CDVD 040				GIC=G1%D1+C	
0262	746D	CDVD 041				GOC=G0%D0+C	
0264	F5D9	CDVD 042		036	INSERT	BR IF S3=1	
0266	A6F6	CDVD 043		046	SHIFT	BAL	
0268	1002	CDVD 044				RST S K=90	
026A	825C	CDVD 045		038	COMPUT	BR	
26F6	2C1B	CDVD 046	SHIFT			PO=P0+K01	
26F8	611D	CDVD 047	SHIFTA			UIC=U1+U1+C	SHIFT
26FA	60CD	CDVD 048				UOC=U0+U0+C	LEFT
26FC	633D	CDVD 049				VIC=V1+V1+C	ROUTINE
26FE	622D	CDVD 050				VOC=V0+V0+C	
2700	EC0D	CDVD 051		057	CYCLE	BR IF P02=1	
2702	699D	CDVD 052				IIC=I1+I1+C	
2704	688D	CDVD 053				IOC=I0+I0+C	
2706	655D	CDVD 054				GIC=G1+G1+C	
2708	644D	CDVD 055				GOC=G0+G0+C	
270A	128E	CDVD 056				RTN	
270C	C418	CDVD 057	CYCLE	063	ARNQ	BR IF G00=0	
270E	1002	CDVD 058				RST S K=90	
2710	79FD	CDVD 059				IIC=I1%H1+C	
2712	78ED	CDVD 060				IOC=I0%H0+C	
2714	757D	CDVD 061				GIC=G1%D1+C	
2716	746D	CDVD 062				GOC=G0%D0+C	
2718	58C9	CDVD 063	ARNQ			PO=T1	
271A	52D9	CDVD 064				P1=V0	
271C	E827	CDVD 065		070	NXTA	BR IF P06=1	REM -
271E	C436	CDVD 066		078	NXTB	BR IF G00=0	REM SHD BE + IS +
2720	97EC	CDVD 067		100	COMR	BAL	
2722	C436	CDVD 068		078	NXTB	BR IF G00=0	
2724	AEFC	CDVD 069	LABELQ	098	DIVCHK	BR	
2726	C42B	CDVD 070	NXTA	072	NXTD	BR IF G00=1	
2728	97EC	CDVD 071		100	COMR	BAL	
272A	F839	CDVD 072	NXTD	079	NXTF	BR IF P07=1	
272C	97F0	CDVD 073	NXIC	102	COMQUO	BAL	
272E	52D9	CDVD 074				P1=V0	
2730	D5BA	CDVD 075		080	FINISH	BR IF S1=0	
2732	CD3B	CDVD 076		080	FINISH	BR IF P1 BIT0=1	
2734	AEFC	CDVD 077		098	DIVCHK	BR	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2736	F82D	CDVD 078	NXTB	073	NXTC	BR IF P07=1	
2738	CD25	CDVD 079	NXTF	069	LABELQ	BR IF P10=1	
273A	58BB	CDVD 080	FINISH			T1=T1H	
273C	72A8	CDVD 081				STH V AS,T+2	
273E	70AA	CDVD 082				STH U AS,T-2	
2740	1B15	CDVD 083				T1=T1*-K10	
2742	74A8	CDVD 084				STH G AS,T+2	
2744	78AA	CDVD 085				STH I AS,T-2	
2746	9256	CDVD 086		CCOM 065	LRSTRA	BAL	
2748	A8DE	CDVD 087		CICY 004	CHECK	BR	
17D8	2002	CDVD 088	COMDVD			SET S K=10	COMP
17DA	711B	CDVD 089				UIC=0-U1+C	DVD
17DC	700B	CDVD 090				UOC=0-U0+C	
17DE	733B	CDVD 091				VIC=0-V1+C	
17E0	722B	CDVD 092				VOC=0-V0+C	
17E2	799B	CDVD 093	CMDVA			IIC=0-I1+C	
17E4	788B	CDVD 094				IOC=0-I0+C	
17E6	755B	CDVD 095				GIC=0-G1+C	
17E8	744B	CDVD 096				GOC=0-G0+C	
17EA	128E	CDVD 097				RTN	
2EFC	9256	CDVD 098	DIVCHK	CCOM 065	LRSTRA	BAL	
2EFE	AB62	CDVD 099		CNVR 120	LAB14	BR	GO FLAG DIVIDE CHECK AND INTRPT
17EC	2002	CDVD 100	COMR			SET S K=10	
17EE	F5E3	CDVD 101		093	CMDVA	BR IF S3=1	
17F0	0008	CDVD 102	COMQUO			RST S1	
17F2	3002	CDVD 103				SET S K=90	
17F4	711B	CDVD 104				UIC=0-U1+C	
17F6	700B	CDVD 105				UOC=0-U0+C	
17F8	733B	CDVD 106				VIC=0-V1+C	
17FA	722B	CDVD 107				VOC=0-V0+C	
17FC	128E	CDVD 108				RTN	

 * CROSS REFERENCE FOR CSECT CDVD *

CDVD 002	CICY 160	
CDVD 004	CDVD 002	
CDVD 006	CICY 192	
CDVD 009	CDVD 005	
CDVD 011	CDVD 009	
CDVD 020	CDVD 017	
CDVD 023	CDVD 020	
CDVD 032	CDVD 030	
CDVD 036	CDVD 042	
CDVD 037	CDVD 035	
CDVD 038	CDVD 045	
CDVD 046	CDVD 036	CDVD 043
CDVD 047	CDVD 034	
CDVD 057	CDVD 051	
CDVD 063	CDVD 057	
CDVD 069	CDVD 079	
CDVD 070	CDVD 065	
CDVD 072	CDVD 070	
CDVD 073	CDVD 078	

* CROSS REFERENCE FOR CSECT CDVD *

CDVD 078	CDVD 066	CDVD 068
CDVD 079	CDVD 072	
CDVD 080	CDVD 075	CDVD 076
CDVD 088	CDVD 021	
CDVD 093	CDVD 101	
CDVD 098	CDVD 031	CDVD 069 CDVD 077
CDVD 100	CDVD 067	CDVD 071
CDVD 102	CDVD 073	

CFAD DESCRIPTIVE TEXT

ENTRY IS MADE TO THE -CFAD- ROUTINE FROM FLOATING POINT I-CYCLES -CFCY-. THE NORMALIZED AND UNNORMALIZED ADD OR SUBTRACT, AND COMPARE OP CODES LISTED BELOW ARE HANDLED BY THIS ROUTINE.

```

*****
*   OP CODE TYPE   * BYTE 1 * BYTE 2 * BYTE 3 * BYTE 4 *
*****
* RR COMPARE (LONG) * 29   * R1  R2 *      *      *
*-----*
* RR ADD NORM. (LONG) * 2A   * R1  R2 *      *      *
*-----*
* RR SUB. NORM. (LONG) * 2B   * R1  R2 *      *      *
*-----*
* RR ADD UNORM. (LONG) * 2E   * R1  R2 *      *      *
*-----*
* RR SUB UNORM. (LONG) * 2F   * R1  R2 *      *      *
*-----*
* RR COMPARE (SHORT) * 39   * R1  R2 *      *      *
*-----*
* RR ADD NORM. (SHORT) * 3A   * R1  R2 *      *      *
*-----*
* RR SUB NORM. (SHORT) * 3B   * R1  R2 *      *      *
*-----*
* RR ADD UNORM. (SHORT) * 3E   * R1  R2 *      *      *
*-----*
* RR SUB UNORM. (SHORT) * 3F   * R1  R2 *      *      *
*-----*
* RX COMPARE (LONG) * 69   * R1  X2 * B2  D2 *      *
*-----*
* RX ADD NORM. (LONG) * 6A   * R1  X2 * B2  D2 *      *
*-----*
* RX SUB NORM. (LONG) * 6B   * R1  X2 * B2  D2 *      *
*-----*
* RX ADD UNORM. (LONG) * 6E   * R1  X2 * B2  D2 *      *
*-----*
* RX SUB UNORM. (LONG) * 6F   * R1  X2 * B2  D2 *      *
*-----*
* RX COMPARE (SHORT) * 79   * R1  X2 * B2  D2 *      *
*-----*
* RX ADD NORM. (SHORT) * 7A   * R2  X2 * B2  D2 *      *
*-----*
* RX SUB NORM. (SHORT) * 7B   * R2  X2 * B2  D2 *      *
*-----*
* RX ADD UNORM. (SHORT) * 7E   * R1  X2 * B2  D2 *      *
*-----*
* RX SUB UNORM. (SHORT) * 7F   * R1  X2 * B2  D2 *      *
*****
    
```

OBJECTIVES

COMPARE OPS - THE FIRST OPERAND IS COMPARED WITH THE SECOND OPERAND. THE CONDITION CODE IS SET TO INDICATE THE RESULT OF THE COMPARE. COMPARISON TAKES INTO ACCOUNT THE SIGN, FRACTION, AND EXPONENT OF EACH OPERAND.

ADD OPS - THE SECOND OPERAND IS ADDED TO THE FIRST OPERAND, AND THE SUM IS PLACED IN THE FIRST OPERAND LOCATION. FOR SHORT OP FORMATS, THE LOW ORDER HALVES OF THE FLOATING POINT REGISTERS ARE NOT USED.

CHARACTERISTICS ARE ALIGNED BEFORE ADDITION TAKES PLACE.

SUBTRACT OPS- THE SECOND OPERAND IS SUBTRACTED FROM THE FIRST OPERAND, AND THE DIFFERENCE IS PLACED IN THE FIRST OPERAND LOCATION. FOR SHORT OP FORMATS THE LOW ORDER HALVES OF THE FLOATING POINT REGISTERS ARE NOT USED.

CHARACTERISTICS ARE ALIGNED BEFORE SUBTRACTION TAKES PLACE.

CONDITION CODES

COMPARE	0=OPERANDS ARE EQUAL 1=FIRST OPERAND IS LOW 2=FIRST OPERAND IS HIGH
ADD AND SUBTRACT	0=RESULT FRACTION IS ZERO 1=RESULT IS LESS THAN ZERO 2=RESULT IS GREATER THAN ZERO 3=RESULT EXPONENT OVERFLOWS

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFAD 001	T			ADD,SUBTRACT,COMPARE	FLOATING POINT
		CFAD 002	*				
		CFAD 003	*				
		CFAD 004	*				
		CFAD 005	*				
		CFAD 006	ASEQ	AL07=72			
0972	3B35	CFAD 007	SCAD			T1=T1\$K30 ** T=ADD OF SAVE AREA-38 OR 78 IN AUX	
0974	9AD0	CFAD 008		CFCO 078	STORE2	BAL	SAVE 2ND OPERAND
0976	5E79	CFAD 009				D1=HO	SAVE R2 EXPONENT
0978	5AE2	CFAD 010				RDH T CA,BC	GET R1 ADDRESS
097A	8212	CFAD 011		CFCO 090	RDOPER	BAL	GET R1 OPERAND
097C	2C07	CFAD 012				PO=0	
097E	5E69	CFAD 013				DO=HO	SAVE R1 EXP
0980	5ED9	CFAD 014				P1=HO	SAVE R1 EXP FOR ORIG SIGN
0982	2040	CFAD 015				SET S5 ** S5	WILL BE RST IF R1 MUST BE RIGHT SHIFT
0984	7671	CFAD 016				DO=DO-D1+1	SUBTRACT R1EXP-R2EXP
		CFAD 017	*				NO 1BC IND. R1 MUST SHIFT DO HAS COUNT
0986	E48E	CFAD 018		022	NOTOVF	BR IF NOVFL	
0988	F491	CFAD 019		023	NOT1BC	BR IF AC=1	OVFL IS ON
098A	56C9	CFAD 020	HAV1BC			PO=DO ** SHIFT COUNT TRUE FORM TO PO	
098C	8956	CFAD 021		027	TCCTRL	BR	
098E	F48B	CFAD 022	NOTOVF	020	HAV1BC	BR IF AC=1	
0990	0040	CFAD 023	NOT1BC			RST S5 ** NO 1BC R1 IS SMALL-MUST SHIFT	
		CFAD 024	*				R2 IS RES EXP MOVE R2 EXP TO R1 OPERAND
0992	57E9	CFAD 025				HO=D1	
0994	7C61	CFAD 026				PO=PO-DO+1 ** SHIFT COUNT IS IN COMP. MUST RECOMP	
0996	F01A	CFAD 027	TCCTRL	029	THISAD	BR IF G07=0	BR IF ADD COMP CTRL IS OK
0998	1C8D	CFAD 028				PO=PO\$K80	SUB,COMP-MUST INVERT T/C BIT
099A	CC1E	CFAD 029	THISAD	031	TRUE	BR IF P00=0	BR IF TRUE
099C	3002	CFAD 030				SET S K=90 ** SET	COMP CTRL AND CARRY
099E	1C85	CFAD 031	TRUE			PO=PO*-K80	REMOVE SIGN FROM SHIFT COUNT
09A0	ECA5	CFAD 032		034	SPEEDY	BR IF HZ=0	BR IF NO HI COUNT ALIGNMENT
09A2	2CF3	CFAD 033				PO=0\$K0F	USE COUNT OF 15 TO ZERO OUT REG
09A4	F0BF	CFAD 034	SPEEDY	047	NOSHIF	BR IF LZ=0	BR IF EXPONENTS ARE EQUAL
09A6	D1B6	CFAD 035		043	SHIFR1	BR IF S5=0	BR IF R1 MUST SHIFT
09A8	3B35	CFAD 036				T1=T1\$K30	T= R2 SAVE
09AA	8212	CFAD 037		CFCO 090	RDOPER	BAL	GET R2
09AC	A656	CFAD 038		CFCO 007	SHIFR4	BAL	
09AE	9AD0	CFAD 039		CFCO 078	STORE2	BAL	PUT SHIFTED R2 BACK IN SAVE
09B0	5AE2	CFAD 040				RDH T DA,BC	GET R1 ADD T=00X8
09B2	8212	CFAD 041		CFCO 090	RDOPER	BAL	
09B4	89C0	CFAD 042		048	ADD	BR	
09B6	1E85	CFAD 043	SHIFR1			HO=HO*-K80	MAKE RESULT SIGN +
09B8	CD3C	CFAD 044		046	DESADD	BR IF P10=0	BR IF ORIG SIGN + SIGN IS OK
09BA	3E85	CFAD 045				HO=HO\$K80	ORIG WAS - MAKE RES -
09BC	A656	CFAD 046	DESADD	CFCO 007	SHIFR4	BAL	GO SHIFT R1 RIGHT
09BE	0040	CFAD 047	NOSHIF			RST S5 ** NO SHIFT	RST S5 FOR GUARD DIGIT CTRL
09C0	3B35	CFAD 048	ADD			T1=T1\$K30	T= 38 OR 78 =R2 SAVE
09C2	0008	CFAD 049				RST S1	FOR GUARD SIGNIF. TEST
09C4	D1CB	CFAD 050		053	R2GUAR	BR IF S5=1	BR IF GUARD IS R2
09C6	75AD	CFAD 051	R2TRUE			G1C=G1\$T0+C	DO SIG TEST ON R1
09C8	89CE	CFAD 052		055	CHKNOR	BR	
09CA	C5C6	CFAD 053	R2GUAR	051	R2TRUE	BR IF S0=0	D TRUE ADD

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
09CC	755B	CFAD 054				G1C=0-G1+C	R2 GUARD COMP
09CE	D052	CFAD 055	CHKNOR	057	NORM	BR IF G05=0	
09D0	0008	CFAD 056				RST S1 ** UNNORMALIZED-GUARD NOT TESTED FOR SIGNIFI.	
09D2	A704	CFAD 057	NORM	CFCO 120	COMPUT	BAL	GO ADD,S,C R1R2
09D4	5AE2	CFAD 058				RDH T DA,8C	GET R1 ADDRESS
09D6	C5EE	CFAD 059		071	TCCMPU	BR IF S0=0	BR IF TRUE COMPUTE
09D8	F50E	CFAD 060		063	NOCARY	BR IF S3=0	COMPLIMENT -CHECK CARRY
09DA	0002	CFAD 061				RST S3	
09DC	89EE	CFAD 062		071	TCCMPU	BR	ALL OK RESULT IS TRUE FORM
09DE	1E8D	CFAD 063	NOCARY			H0=H0K80	INV SIGN-COMP OP -MUST RECOMP
09E0	E070	CFAD 064		072	CPFIN	BR IF G06=0	BR IF COMPARE
09E2	0008	CFAD 065				RST S1	RST S1 FOR RES CHECK OF COMP
09E4	2002	CFAD 066				SET S3	COMP GUARD
09E6	755B	CFAD 067				G1C=0-G1+C	
09E8	D06C	CFAD 068		070	SAVSIG	BR IF G05=0	BR IF NORMAL. SAVE GUARD SIGNIF.
09EA	00C8	CFAD 069				RST S1	UNORM. RST SIG. TEST BIT
09EC	9364	CFAD 070	SAVSIG	CFCO 103	RECOMP	BAL	
09EE	E07F	CFAD 071	TCCMPU		080	NOTCP	BR IF G06=1
09F0	9258	CFAD 072	CPFIN	CCOM 066	LRSTRB	BAL	BR IF ADD OR SUB
09F2	D5F9	CFAD 073			077	COMNZ	SET
09F4	F5F9	CFAD 074			077	COMNZ	BR IF OVERFLOW ON CMP.--NOT-0
09F6	A8DC	CFAD 075		CICY 003	CHECKX	BR ** RES = 0	COND
		CFAD 076	*				CODE
09F8	CE7D	CFAD 077	COMNZ	079	COMPNE	BR IF H00=1	BR IF NEG
09FA	ACA4	CFAD 078		CLOG 052	OP95A	BR	SET CC 2
09FC	A91A	CFAD 079	COMPNE	CLOG 082	SETNZ	BR	SET CC 1
09FE	F5A9	CFAD 080	NOTCP	104	FRACOV	BR IF S3=1	BR IF CARR RES NOT 0
		CFAD 081	*				FRACTION MUST SHIFT RIGHT 4
0A00	5CC2	CFAD 082				RDH P DA,8B	RESTORE P
0A02	D5CA	CFAD 083		123	FRACZO	BR IF S1=0	GO CHECK SIG MASK
0A04	D043	CFAD 084		118	UNOREN	BR IF G05=1	
0A06	0FF3	CFAD 085	CONORM			Z=H1*-KOF	CHECK HI DIGIT FOR NORMALIZING
0A08	C4C2	CFAD 086		118	UNOREN	BR IF ZNZ	BR IF NO NORMALIZE REQUIRED
0A0A	A1E4	CFAD 087		CFCO 034	SHIFL4	BAL	
0A0C	2EFF	CFAD 088				H0=H0+KFF	DECREMENT EXPONENT
0A0E	E456	CFAD 089		093	NOOVER	BR IF NOVFL	
0A10	F499	CFAD 090		094	UNDERF	BR IF AC=1	
0A12	2507	CFAD 091	HA1BC			G1=0 ** ZERO GUARD-HAVE 1BC NO UNDERFLOW	
0A14	8A06	CFAD 092		085	CONORM	BR	
0A16	F493	CFAD 093	NOOVER	091	HA1BC	BR IF AC=1	
0A18	ED1F	CFAD 094	UNDERF	099	TAKEIN	BR IF P12=1	EXPONENT HAS UNDERFLOWED
		CFAD 095	*				IF UNDF IS MASKED RESULT IS 0'D OUT
		CFAD 096	*				IF NOT MASKED CONT NORM. BUT TAKE INTER.
0A1A	2C07	CFAD 097				P0=0	SET CC0
0A1C	B260	CFAD 098		CFHA 029	DIVNOI	BR ** UNDERFLOW BUT NO INT,GO STORE 0,REST I,GO ICYC	
0A1E	2080	CFAD 099	TAKEIN			SET S4	REMEMBER INT
0A20	1E8D	CFAD 100				H0=H0K80	INVERT SIGN
0A22	27D3	CFAD 101				D1=0\$KOD	UNDERFLOW INT CODE
0A24	25C7	CFAD 102				G1=0	ZERO OUT GUARD
0A26	8A06	CFAD 103		085	CONORM	BR	CONTINUE NORMALIZING
0A28	2A13	CFAD 104	FRACOV			T0=0\$K01	FRAC.HAS OVERF. INSERT 01 FOR
		CFAD 105	*				RIGHT SHIFT
0A2A	2C13	CFAD 106				P0=0\$K01	SET SHIFT COUNT =1

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0A2C	A656	CFAD 107		CFCO 007	SHIFR4	BAL	
0A2E	5CC2	CFAD 108				RDH P DA,B8	RESTORE P
0A30	2A07	CFAD 109				TO=0	
0A32	2E1B	CFAD 110				HO=HO+K01	INC EXPONENT
0A34	E48B	CFAD 111		114	ACBRAN	BR IF OVFL	CHECK
0A36	F48D	CFAD 112		115	EXPOV	BR IF AC=1	FOR EXPONENT OVERFLOW
0A38	8A42	CFAD 113	NO1BC	118	UNOREN	BR ** NO 1BC-NO OVERFLOW-GO SET CC+STORE RESULT	
0A3A	F489	CFAD 114	ACBRAN	113	NO1BC	BR IF AC=1	
0A3C	1E8D	CFAD 115	EXPOV			HO=HO+K80	INVERT SIGN
0A3E	27C3	CFAD 116				DI=0\$K0C	SET OVERF INT CODE
0A40	2C80	CFAD 117				SET S4	REMEMBER INT TO BE TAKEN
0A42	2C25	CFAD 118	UNOREN			PO=0\$K20	SET CC 2 +
0A44	CE48	CFAD 119		121	FIN	BR IF H00=0	IF POS CC IS CORRECT
0A46	1C7D	CFAD 120				PO=PO+K70	RES - CORRECT CC = 1
0A48	B266	CFAD 121	FIN	CFHA 032	STORE1	BR	STORE RESULT,RESTORE I , GO TO
		CFAD 122	*				I CYCLES OR INTER. IF S4=1
0A4A	FD4F	CFAD 123	FRACZO	125	SIGINT	BR IF P13=1	BR IF SIG INT NOT MASKED
0A4C	2E07	CFAD 124				HO=0	ZERO OUT EXP -NO INT
0A4E	1E85	CFAD 125	SIGINT			HO=HO*-K80	MAKE TRUE ZERO
0A50	9AD0	CFAD 126		CFCO 078	STORE2	BAL	STORE 0 RES
0A52	925A	CFAD 127		CCOM 067	LRSTRC	BAL	RESTORE I
0A54	2C07	CFAD 128				PO=0	C C O
0A56	FD58	CFAD 129		131	TSIGIN	BR IF P13=1	
0A58	A8DE	CFAD 130		CICY 004	CHECK	BR	
0A5A	2BE3	CFAD 131	TSIGIN			T1=0\$K0E	
0A5C	9C70	CFAD 132		BPSW 003	PROG	BR	
		CFAD 133	AEND				

 * CROSS REFERENCE FOR CSECT CFAD *

CFAD 007	CFCY 061	CFCY 062	CFCY 063	CFCY 066	CFCY 067
CFAD 020	CFAD 022				
CFAD 022	CFAD 018				
CFAD 023	CFAD 019				
CFAD 027	CFAD 021				
CFAD 029	CFAD 027				
CFAD 031	CFAD 029				
CFAD 034	CFAD 032				
CFAD 043	CFAD 035				
CFAD 046	CFAD 044				
CFAD 047	CFAD 034				
CFAD 048	CFAD 042				
CFAD 051	CFAD 053				
CFAD 053	CFAD 050				
CFAD 055	CFAD 052				
CFAD 057	CFAD 055				
CFAD 063	CFAD 060				
CFAD 070	CFAD 068				
CFAD 071	CFAD 059	CFAD 062			
CFAD 072	CFAD 064				
CFAD 077	CFAD 073	CFAD 074			
CFAD 079	CFAD 077				
CFAD 080	CFAD 071				

* CROSS REFERENCE FOR CSECT CFAD *

CFAD 085	CFAD 092	CFAD 103
CFAD 091	CFAD 093	
CFAD 093	CFAD 089	
CFAD 094	CFAD 090	
CFAD 099	CFAD 094	
CFAD 104	CFAD 080	
CFAD 113	CFAD 114	
CFAD 114	CFAD 111	
CFAD 115	CFAD 112	
CFAD 118	CFAD 084	CFAD 086 CFAD 113
CFAD 121	CFAD 119	
CFAD 123	CFAD 083	
CFAD 125	CFAD 123	
CFAD 131	CFAD 129	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFCO 001	T			*** COMMON FLOATING POINT ROUTINES ***	
		CFCO 002	*				
		CFCO 003	*				
		CFCO 004	*			*** SHIFT RIGHT 4 ***	
		CFCO 005	*				
		CFCO 006	*				
2656	F45C	CFCO 007	SHIFR4	010	RLONG	BR IF G03=0	BR IF LONG
2658	4953	CFCO 008				G1=I1XH+G1L	
265A	F46F	CFCO 009		019	RSHORT	BR IF G03=1	UNC BRANCH TO SHORT OP
265C	4153	CFCO 010	RLONG			G1=U1XH+G1L	SAVE GUARD LONG
265E	5115	CFCO 011				U1=U1XL	
2660	4013	CFCO 012				U1=U0XH+U1L	
2662	5005	CFCO 013				U0=U0XL	
2664	4303	CFCO 014				U0=V1XH+U0L	
2666	5335	CFCO 015				V1=V1XL	
2668	4233	CFCO 016				V1=V0XH+V1L	
266A	5225	CFCO 017				V0=V0XL	
266C	4923	CFCO 018				V0=I1XH+V0L	
266E	5995	CFCO 019	RSHORT			I1=I1XL	
2670	4893	CFCO 020				I1=I0XH+I1L	
2672	5885	CFCO 021				I0=I0XL	
2674	4F83	CFCO 022				I0=H1XH+I0L	
2676	5FF5	CFCO 023				H1=H1XL	
2678	4AF3	CFCO 024				H1=TOXH+H1L	
267A	2CFF	CFCO 025				P0=P0+KFF	DEC SHIFTCOUNT
267C	C4D6	CFCO 026		007	SHIFR4	BR IF ZNZ	
267E	128E	CFCO 027				RTN	
		CFCO 028	*				
		CFCO 029	*				
		CFCO 030	*				
		CFCO 031	*			*** SHIFT LEFT 4 ***	
		CFCO 032	*				
		CFCO 033	*				
21E4	5FF3	CFCO 034	SHIFL4			H1=H1XH	
21E6	48F5	CFCO 035				H1=I0XL\$H1H	
21E8	5883	CFCO 036				I0=I0XH	
21EA	4985	CFCO 037				I0=I1XL\$I0H	
21EC	5993	CFCO 038				I1=I1XH	
21EE	F474	CFCO 039		042	LONGL	BR IF G03=0	BR IF LONG
21F0	4595	CFCO 040				I1=G1XL\$I1H	SHIFT IN GUARD
21F2	128E	CFCO 041				RTN	
21F4	4295	CFCO 042	LONGL			I1=V0XL\$I1H	
21F6	5223	CFCO 043				V0=V0XH	
21F8	4325	CFCO 044				V0=V1XL\$V0H	
21FA	5333	CFCO 045				V1=V1XH	
21FC	4035	CFCO 046				V1=U0XL\$V1H	
21FE	5003	CFCO 047				U0=U0XH	
2200	4105	CFCO 048				U0=U1XL\$U0H	
2202	5113	CFCO 049				U1=U1XH	
2204	4515	CFCO 050				U1=G1XL\$U1H	SHIFT IN LONG GUARD
2206	128E	CFCO 051				RTN	
		CFCO 052	*				
		CFCO 053	*				

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFCO 054	*				
		CFCO 055	*				
0BDA	0008	CFCO 056	ZTEST			*** ZERO TEST ***	
0BDC	6FF5	CFCO 057				RST S1	
0BDE	4886	CFCO 058				H1=H1&H1	MOVE THRU ALU -SET S1 IF NZ
0BE0	F467	CFCO 059		062	ZSHORT	I=I	
0BE2	4226	CFCO 060				BR IF G03=1	BR IF SHORT
0BE4	4006	CFCO 061				V=V	
0BE6	128E	CFCO 062	ZSHORT			U=U	
		CFCO 063	*			RTN	
		CFCO 064	*				
		CFCO 065	*				
		CFCO 066	*				
		CFCO 067	*				
		CFCO 068	*			*** ZERO OUT RESULT ***	
0136	4E52	CFCO 069	ZEROUT			RDH H DC,9A	READ ZERO CONSTANT INTO H
		CFCO 070	*				
0138	48E6	CFCO 071				I=H	
013A	42E6	CFCO 072				V=H	
013C	40E6	CFCO 073				U=H	
013E	128E	CFCO 074				RTN	
		CFCO 075	*				
		CFCO 076	*				
		CFCO 077	*			*** STORE ROUTINE ***	
1AD0	7EA8	CFCO 078	STORE2			STH H AS,T+2	
1AD2	78A8	CFCO 079				STH I AS,T+2	
1AD4	F45B	CFCO 080		083	SHORTS	BR IF G03=1	BR IF SHORT
1AD6	72A8	CFCO 081				STH V AS,T+2	
1AD8	70A0	CFCO 082				STH U AS,T+0	
1ADA	1B73	CFCO 083	SHORTS			T1=T1*-K07	RST T TO 00X8
1ADC	128E	CFCO 084				RTN	
		CFCO 085	*				
		CFCO 086	*				
		CFCO 087	*				
		CFCO 088	*				
		CFCO 089	*			*** READ REGISTER ***	
0212	5EA8	CFCO 090	RDOPER			RDH H AS,T+2	
0214	58A8	CFCO 091				RDH I AS,T+2	
0216	F41D	CFCO 092		095	SHORTR	BR IF G03=1	BR IF SHORT
0218	52A8	CFCO 093				RDH V AS,T+2	
021A	50A0	CFCO 094				RDH U AS,T+0	
021C	1B73	CFCO 095	SHORTR			T1=T1*-K07	T1= X8
021E	128E	CFCO 096				RTN	
		CFCO 097	*				
		CFCO 098	*				
		CFCO 099	*				
		CFCO 100	*			*** COMPLIMENT ROUTINE ***	
		CFCO 101	*				
		CFCO 102	*				
1364	F46F	CFCO 103	RECOMP	108	SHORTC	BR IF G03=1	BR IF SHORT
1366	711B	CFCO 104				U1C=0-U1+C	
1368	700B	CFCO 105				U0C=0-U0+C	
136A	733B	CFCO 106				V1C=0-V1+C	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
136C	7228	CFCO 107				VOC=0-V0+C	
136E	7998	CFCO 108	SHORTC			IIC=0-I1+C	
1370	7888	CFCO 109				IOC=0-I0+C	
1372	7FF8	CFCO 110				HIC=0-H1+C	
1374	EC79	CFCO 111		113	COMPAR	BR IF G06=1	BR IF NOT COMPARE OP
1376	0002	CFCO 112				RST S3	
1378	128E	CFCO 113	COMPAR			RTN	
		CFCO 114	*				
		CFCO 115	*				
		CFCO 116	*				
		CFCO 117	*			*** ADD AND DIVIDE COMUTE R1-R2 ***	
		CFCO 118	*				
		CFCO 119	*				
27D4	3BE3	CFCO 120	COMPUT			T1=T1\$KOE	SET T TO LOW HALF 00XE
27D6	F45C	CFCO 121		124	LOCOMP	BR IF G03=0	BR IF LONG
27D8	1B43	CFCO 122				T1=T1*-K04	T1 = 00XA FOR SHORT LOW
27DA	F469	CFCO 123		130	SHCCMP	BR IF G03=1	UNC BRANCH
27DC	56AA	CFCO 124	LOCOMP			RDH D AS,T-2	T = 00XE T IS SET
27DE	717D	CFCO 125				UIC=U1%D1+C	TO 38 OR 78
27E0	706D	CFCO 126				UOC=U0%D0+C	WHICH IS A
27E2	56AA	CFCO 127				RDH D AS,T-2	T = 00XC SAVE AREA
27E4	737D	CFCO 128				VIC=V1%D1+C	FOR R2 OP
27E6	726D	CFCO 129				VOC=V0%D0+C	
27E8	56AA	CFCO 130	SHCOMP			RDH D AS,T-2	T = 00XA
27EA	797D	CFCO 131				IIC=I1%D1+C	
27EC	786D	CFCO 132				IOC=I0%D0+C	
27EE	56A0	CFCO 133				RDH D AS,T+0	T = 00X8
27F0	7F7D	CFCO 134				HIC=H1%D1+C	
27F2	C1F7	CFCO 135		137	DIVCOM	BR IF S4=1	BR IF DIVIDE
27F4	128E	CFCO 136				RTN	A, S, C
27F6	7E6D	CFCO 137	DIVCOM			HOC=H0%D0+C	
27F8	128E	CFCO 138				RTN	
		CFCO 139	*				
		CFCO 140	*				
		CFCO 141	*			*** RX OPERAND FETCH ***	
		CFCO 142	*				
2574	5E18	CFCO 143	RXOPER			RDH H U+2	
2576	5818	CFCO 144				RDH I U+2	
2578	F47F	CFCO 145		148	XSHORT	BR IF G03=1	
257A	5218	CFCO 146				RDH V U+2	
257C	5010	CFCO 147				RDH U U+0	
257E	128E	CFCO 148	XSHORT			RTN	

 * CROSS REFERENCE FOR CSECT CFCO *

CFCO 007	CFAD 038	CFAD 046	CFAD 107	CFCO 026	CFDV 016	CFHA 010
CFCO 010	CFCO 0C7					
CFCO 019	CFCO 0C9					
CFCO 034	CFAD 087	CFHA 016	CFMD 030	CFMU 046		
CFCO 042	CFCO 039					
CFCO 056	CFDV 048	CFLS 025	CFMD 010	CFMU 051		
CFCO 062	CFCO 059					
CFCO 069	CFHA 030					

CFCY DESCRIPTIVE TEXT

THE FLOATING POINT I-CYCLES ROUTINE -CFCY- IS ENTERED AT LABELS -RROP- OR -RXOP- FROM THE MAIN I-CYCLES ROUTINE -ICYC-.

THE CHECK FOR CORRECT OPERAND ADDRESS BOUNDRIES AND THE CHECK FOR VALID FLOATING POINT REGISTER ADDRESSES IS MADE.

THE SECOND OPERAND IS FETCHED AND THE FIRST OPERAND ADDRESS IS DEVELOPED BEFORE A BRANCH IS TAKEN TO THE INDIVIDUAL FLOATING POINT OPERATION ROUTINES.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFCY 001	T			FLOATING POINT I CYCLE DECODE RR AND RX	
		CFCY 003	*			*	
		CFCY 004	*			ALL OPS EXCEPT STORE BEGIN ALIKE AS FOLLOWS	
		CFCY 005	*			2ND OPERAND IS IN LS HO,H1 IO,I1,G1 (SHORT)	
		CFCY 006	*			HO,H1,IO,I1,VO,V1,UO,U1,G1 (LONG)	
		CFCY 007	*			1ST OPERAND-NORM.OR UNNOR. IS IN R1 LOCATION	
		CFCY 008	*			*** RR I CYCLES FLTPT ***	
2844	E1C9	CFCY 012	RROP	014	NOEXEC	BR IF S6=1	CHK EXECUTE STAT BR IF NOT
2846	3485	CFCY 013				GO=GO\$K80	REMEMBER EXECUTE
2848	0567	CFCY 014	NOEXEC			Z=G1*-K66	CHEC FOR ODD REG=SPEC CHECK
284A	C4C2	CFCY 015		051	SPEC	BR IF ZNZ	
284C	8DC2	CFCY 016		CCOM 057	LSAVEA	BAL	SAVE P,I,G
284E	5583	CFCY 017				T1=G1XH	R2 ADDRESS
2850	3883	CFCY 018				T1=T1\$K08	
2852	8212	CFCY 019		CFCO 090	RDOPER	BAL	GET 2ND OP- HOH1IO11 (SHORT)
		CFCY 020	*				HOH1IO11VOV1UOUI (LONG)
2854	458B	CFCY 021	R1OPER			T1=G1H+T1L	R1 ADDRESS=T1 X8
2856	25C7	CFCY 022				G1=0	
2858	7AE2	CFCY 023				STH T DA,BC	SAVE R1 OPER ADD IN AUX
285A	10CE	CFCY 024				RST S K=FC	
285C	8CC1	CFCY 025		052	FLOAT N	N=GOL	SEP. OPS
		CFCY 026	*			*****	
		CFCY 031	*			*** RX I CYCLES FLTPT ***	
		CFCY 032	*				
		CFCY 033	*				
2820	AF4E	CFCY 034	RXOP	CICY 081	BASE2	BAL	GET REST OF OP
2822	2B83	CFCY 035				T1=0\$K08	
2824	8DC2	CFCY 036		CCOM 057	LSAVEA	BAL	SAVE P,I,G
2826	0565	CFCY 037				Z=G1*-K60	CHK IF R2 NOT 0,2,4,6
2828	E0C2	CFCY 038		051	SPEC	BR IF HZNZ	
282A	AB53	CFCY 039		CICY 045	ADDERR N	BR IF H1=NZ	CHK 2ND OPER ADD TOO LARGE
282C	F43F	CFCY 040		049	SHORT	BR IF G03=1	BR IF SHORT OP
282E	0183	CFCY 041				Z=U1*-K08	CHK FOR DOUBLE BOUNDARY
2830	F0C2	CFCY 042		051	SPEC	BR IF LZNZ	
2832	C038	CFCY 043	LOCKLD	046	STORCK	BR IF G04=0	
2834	A574	CFCY 044		CFCO 143	RXOPER	BAL	
2836	A854	CFCY 045		021	R1OPER	BR	
2838	04F5	CFCY 046	STORCK			Z=GO*-KFO	LOOK FOR 60 OR 70 OP

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
283A	FG8A	CFCY 047		057	FLOAT 5	BR IF LZNZ	OPER NOT STORE OP
283C	8D6C	CFCY 048		CFLS 034	STORE	BR	
283E	01C3	CFCY 049	SHORT			Z=U1*-KOC	CHK WORD BOUNDARY
2840	F0B3	CFCY 050		043	LOOKLD	BR IF LZ=0	
2842	AE98	CFCY 051	SPEC	CICY 059	SPECHK	BR	
2800	B384	CFCY 052	FLOAT 0	CFLS 007	LOADP	BR	LOAD POS
2802	A150	CFCY 053	FLOAT 1	CFLS 014	LOADN	BR	LOAD NEG
2804	A154	CFCY 054	FLOAT 2	CFLS 022	LOADT	BR	LOAD AND TEST
2806	A152	CFCY 055	FLOAT 3	CFLS 021	LOADC	BR	LOAD COMPLIMENT
2808	B23E	CFCY 056	FLOAT 4	CFHA 006	HALVE	BR	HALVE
280A	B376	CFCY 057	FLOAT 5	068	NOGOOD	BR	OP ERROR
280C	B376	CFCY 058	FLOAT 6	068	NOGOOD	BR	OP ERROR
280E	B376	CFCY 059	FLOAT 7	068	NOGOOD	BR	OP ERROR
2810	B266	CFCY 060	FLOAT 8	CFHA 032	STORE1	BR	LOAD
2812	8972	CFCY 061	FLOAT 9	CFAD 007	SCAD	BR	COMPARE-LONG, SHORT-RR OR RX
2814	8972	CFCY 062	FLOAT A	CFAD 007	SCAD	BR	ADD NORMALIZED LONG OR SHORT
2816	8972	CFCY 063	FLOAT B	CFAD 007	SCAD	BR	SUBTRACT NORM LONG OR SHORT
2818	9D80	CFCY 064	FLOAT C	CFMD 010	MULPRE	BR	MULTIPLY LONG OR SHORT
281A	9D7E	CFCY 065	FLOAT D	CFMD 009	DIVPRE	BR	DIVIDE LONG OR SHORT
281C	8972	CFCY 066	FLOAT E	CFAD 007	SCAD	BR	ADD UNNORMALIZED LONG OR SHORT
281E	8972	CFCY 067	FLOAT F	CFAD 007	SCAD	BR	SUBTRACT UNNORMALIZED LONG OR SH
3376	925A	CFCY 068	NOGOOD	CCOM 067	LRSTRC	BAL	RESTORE I
3378	A05A	CFCY 069		CICY 066	OPERR	BR	

 * CROSS REFERENCE FOR CSECT CFCY *

CFCY 012	CICY 018	CICY 019	
CFCY 014	CFCY 012		
CFCY 021	CFCY 045		
CFCY 034	CICY 022	CICY 023	
CFCY 043	CFCY 050		
CFCY 046	CFCY 043		
CFCY 049	CFCY 040		
CFCY 051	CFCY 015	CFCY 038	CFCY 042
CFCY 052	CFCY 025		
CFCY 057	CFCY 047		
CFCY 068	CFCY 057	CFCY 058	CFCY 059

CFDV DESCRIPTIVE TEXT

THE FLOATING POINT DIVIDE INSTRUCTIONS ARE FIRST HANDLED BY THE -CFMD- ROUTINE. THERE THE ZERO FRACTION TEST, PRENORMALIZATION, AND SUBTRACTION OF EXPONENTS IS MADE.

THE -CFDV- ROUTINE HANDLES THE COMPUTATION OF THE QUOTIENT AND THE STORING OF THE RESULTS. THE -CFCD- ROUTINE IS USED FOR ACTUAL COMPUTATION, RIGHT SHIFTING, AND ZERO TESTING OF THE RESULT. THE -CFHA- ROUTINE HANDLES LEFT SHIFTING, INTERRUPT CHECKING, AND THE DIVIDE EXIT TO EITHER THE INTERRUPT OR I-CYCLES ROUTINE.

OP CODES HANDLED BY THE -CFDV- ROUTINE

```
*****
*   OP CODE TYPE       * BYTE 1 * BYTE 2 * BYTE 3 * BYTE 4 *
*****
* RR DIVIDE (LONG)    *   2D  *  R1  R2 *         *
*-----*-----*-----*-----*
* RR DIVIDE (SHORT)   *   3D  *  R1  R2 *         *
*-----*-----*-----*-----*
* RX DIVIDE (LONG)    *   6D  *  R1  X2 * B2 D2 * D2 *
*-----*-----*-----*-----*
* RX DIVIDE (SHORT)   *   7D  *  R1  X2 * B2 D2 * D2 *
*****
```

OBJECTIVE -- THE FIRST OPERAND IS DIVIDED BY THE SECOND OPERAND. THE QUOTIENT REPLACES THE FIRST OPERAND.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFDV 001	T			FLOATING POINT DIVIDE	
		CFDV 002	*				
		CFDV 003	*				
		CFDV 004	*				
		CFDV 005	*				
1F84	3082	CFDV 006	DIVIDE			SET S K=98	SET S0,S3,S4
1F86	A7D4	CFDV 007		CFCD 120	COMPUT	BAL	DO TRIAL SUBTRACT
1F88	5AE2	CFDV 008				RDH T DA,BE	
1F8A	8212	CFDV 009		CFCD 090	RDOPER	BAL	GET R1
1F8C	46A6	CFDV 010				D=T	
1F8E	3835	CFDV 011				T1=T1\$K30	T= R2 SAVE 0038 OR 0078
1F90	3713	CFDV 012				D1=D1\$K01	D=QUOTIENT ADD 00R19
1F92	76F2	CFDV 013				STH D DA,BE	SAVE QUOT ADD
1F94	F5A4	CFDV 014		022	NCNOSH	BR IF S3=0	
1F96	2C13	CFDV 015				PO=0\$K01	R1 MUST SHIFT RIGHT ONE DIGIT
1F98	A656	CFDV 016		CFCD 007	SHIFR4	BAL	
1F9A	5CA2	CFDV 017				RDH P DA,AC	GET RESULT EXPONENT
1F9C	2C1B	CFDV 018				PO=PO+K01	INC EXP
1F9E	C4A2	CFDV 019		021	FIXUND	BR IF ZNZ	CHK FOR PREV. UNDERFLOW
1FA0	1D13	CFDV 020				P1=P1*-K01	UNDERFLOW-CORRECTED RST IND.
1FA2	7CA2	CFDV 021	FIXUND			STH P DA,AC	PUT ADJ. RES. EXPONENT BACK
1FA4	A6D8	CFDV 022	NCNOSH	CFHA 044	DIVIDL	BAL	SHIFT LEFT 1
1FA6	3002	CFDV 023				SET S K=90	SET S0,S3
1FA8	0040	CFDV 024	LOOP1			RST S5	
1FAA	2D13	CFDV 025				P1=0\$K01	SET QUOTIENT FLAG
1FAC	A7D4	CFDV 026	LOOP	CFCD 120	COMPUT	BAL	SUB VISOR FROM DEND
1FAE	F5F0	CFDV 027		034	DOADD	BR IF S3=0	IF NO CARRY - DO AN ADD

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1FB0	A6D8	CFDV 028		CFHA 044	DIVIDL	BAL	SHIFT DEND AND QUOT FLAG LEFT 1
1FB2	3002	CFDV 029				SET S K=90	
1FB4	D1B0	CFDV 030	QUOTCH	037	STQUOT	BR IF S5=1	BR IF QUOTIENT BYTE FULL
1FB6	CD2C	CFDV 031		026	LOOP	BR IF P10=0	CONTINUE COMPUTE IF NO QUOT FLAG
1FB8	2040	CFDV 032				SET S5	7TH TIME-DO 1 MORE THEN STORE QU
1FBA	9FAC	CFDV 033		026	LOOP	BR	
1FF0	A6D8	CFDV 034	DOADD	CFHA 044	DIVIDL	BAL	SHIFT LEFT
1FF2	1002	CFDV 035				RST S K=90	ADD VISOR
1FF4	9FB4	CFDV 036		030	QUOTCH	BR	TO DEND
1FBC	56F2	CFDV 037	STQUOT			RDH D DA, BE	GET QUOT ADDRESS
1FBE	7D68	CFDV 038				STB P1 AS, D+1	STORE QUOT BYTE
1FC0	76F2	CFDV 039				STH D DA, BE	STORE EQUOT ADDRESS
1FC2	F44A	CFDV 040		044	LONGD	BR IF G03=0	
1FC4	07CB	CFDV 041				Z=D1=K0C	CHECK FOR 00XC
1FC6	FOA8	CFDV 042		024	LOOP1	BR IF LZNZ	END OF SHORT
1FC8	9FCE	CFDV 043		046	ENDDIV	BR	SHORT DIV FINISHED
1FCA	5779	CFDV 044	LONGD			D1=D1	
1FCC	FOA8	CFDV 045		024	LOOP1	BR IF LZNZ	BR IF LONG NOT FINISHED
1FCE	5AE2	CFDV 046	ENDDIV			RDH T DA, BC	GE R1 ADDRESS
1FD0	8212	CFDV 047		CFC0 090	RDCPER	BAL	GET QUOTIENT
1FD2	8BDA	CFDV 048		CFC0 056	ZTEST	BAL	
1FD4	D5D9	CFDV 049		051	GETEXP	BR IF S1=1	BR IF QUOT NOT ZERO
1FD6	B260	CFDV 050		CFHA 029	DIVNOI	BR	RESULT IS 0 STORE 0, GET I, GO, ICY
1FD8	56A2	CFDV 051	GETEXP			RDH D DA, AC	DO=EXP-----D1=SIGN+FLAGS
1FDA	5659	CFDV 052				G1=DO	MOVE RES EXP FOR CORREC+STORE
1FDC	5CC2	CFDV 053				RDH P DA, B8	RESTORE MASK
1FDE	1585	CFDV 054				G1=G1*-K80	MAKE RES SIGN +
1FE0	C764	CFDV 055		057	STOEXP	BR IF D10=0	BR IF RES IS +
1FE2	3585	CFDV 056				G1=G1\$K80	SIGN - MAKE RESULT -
1FE4	75A0	CFDV 057	STOEXP			STB G1 AS, T+0	
1FE6	F377	CFDV 058		063	UNDERF	BR IF D17=1	BR IF UNDERFLOW FLAG IS ON
1FE8	C66D	CFDV 059		061	OVFLOW	BR IF D00=1	BR IF OVERFLOW
1FEA	0080	CFDV 060				RST S4	REST INT FLAG IF NO OVERFLOW
1FEC	27C3	CFDV 061	OVFLOW			D1=0\$K0C	
1FEE	B268	CFDV 062		CFHA 033	DIVQVF	BR	GO RESTORE I , TAKE INT OR GO ICY
1FF6	27D3	CFDV 063	UNDERF			D1=0\$K0D	
1FF8	55E9	CFDV 064				HO=G1	MOVE RES EXP FOR UND STORE
1FFA	B25E	CFDV 065		CFHA 028	UNDCHEK	BR	GO CHECK UNDERFLOW MASK

 * CROSS REFERENCE FOR CSECT CFDV *

CFDV 006	CFMD 054	
CFDV 021	CFDV 019	
CFDV 022	CFDV 014	
CFDV 024	CFDV 042	CFDV 045
CFDV 026	CFDV 031	CFDV 033
CFDV 030	CFDV 036	
CFDV 034	CFDV 027	
CFDV 037	CFDV 030	
CFDV 044	CFDV 040	
CFDV 046	CFDV 043	
CFDV 051	CFDV 049	
CFDV 057	CFDV 055	

* CROSS REFERENCE FOR CSECT CFDV *

CFDV 061 CFDV 059
CFDV 063 CFDV 058

CFHA DESCRIPTIVE TEXT

THE -CFHA- ROUTINE HANDLES THE FOLLOWING OP CODES-

RROP 34 (SHORT OPERANDS)
RXOP 24 (LONG OPERANDS)

THE -CFHA- ROUTINE PERFORMS THE FLOATING POINT HALVE OPERATION. THIS IS DONE BY SHIFTING THE SECOND OPERAND FRACTION LEFT 3 BITS THEN RIGHT 4 BITS. THE RESULT IS NORMALIZED AND STORED IN THE FIRST OPERAND LOCATION.

THE -CFHA- ROUTINE PERFORMS THE EXIT TO I-CYCLES OR THE INTERRUPT ROUTINE FOR THE FLOATING POINT LOAD OPS, AND CERTAIN CASES FOR FLOATING DIVIDE, MULTIPLY, AND ADD OPS.

THE -CFHA- ROUTINE ALSO PERFORMS THE LEFT SHIFT OF THE QUOTIENT FOR THE FLOATING DIVIDE OPERATION.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFHA 001	T			*** HALVE OP	*** FLOATING POINT
		CFHA 002	*				
		CFHA 003	*				
		CFHA 004	*				
		CFHA 005	*				
323E	2C13	CFHA 006	HALVE			PO=0\$K01	SET RIGHT SHIFT COUNT
3240	A6DC	CFHA 007		046	LEFT1	BAL	SHIFT
3242	A6DC	CFHA 008		046	LEFT1	BAL	LEFT
3244	A6DC	CFHA 009		046	LEFT1	BAL	ONE THREE TIMES
3246	A656	CFHA 010		CFCO 007	SHIFR4	BAL	RIGHT 1 = LEFT 3 THEN RIGHT 4
3248	5CC2	CFHA 011				RDH P DA,B8	
324A	D5E4	CFHA 012		031	FRAC00	BR IF S1=0	BR IF 2ND OPER WAS 0
324C	2A07	CFHA 013				TO=0	
324E	0FF3	CFHA 014	HNORM			Z=H1*-KOF	CHECK HI DIGIT
3250	C4E6	CFHA 015		032	STORE1	BR IF ZNZ	
3252	A1E4	CFHA 016		CFCO 034	SHIFL4	BAL	GO NORMALIZE
3254	25C7	CFHA 017				G1=0	ZERO OUT GUARD
3256	2EFF	CFHA 018				HO=HO+KFF	DEC EXPONENT
3258	E4F2	CFHA 019		022	NOVFLH	BR IF NOVFL	CHECK
325A	F4F5	CFHA 020		023	HAUND	BR IF AC=1	IBC
325C	B24E	CFHA 021	OK1BC	014	HNORM	BR	FOR
3272	F40D	CFHA 022	NOVFLH	021	OK1BC	BR IF AC=1	UNDERFLOW NO IBC=UNDERF
3274	ED62	CFHA 023	HAUND	030	HZER	BR IF P12=0	BR IF NO INT. TO BE TAKEN
3276	1E8D	CFHA 024				HO=HO+K80	INVERT THE SIGN BIT
3278	27D3	CFHA 025				D1=0\$K0D	INT CODE SAVE
327A	2C80	CFHA 026				SET S4	REMEMBER INT
327C	B24E	CFHA 027		014	HNORM	BR	
325E	ED67	CFHA 028	UNDCHK	032	STORE1	BR IF P12=1	BR IF INT TO BE TAKEN
3260	0080	CFHA 029	DIVNOI			RST S4	
3262	8136	CFHA 030	HZER	CFCO 069	ZEROUT	BAL	
3264	2EC7	CFHA 031	FRAC00			HO=0	ZERO OUT EXPONENT
3266	9AD0	CFHA 032	STORE1	CFCO 078	STORE2	BAL	
3268	925A	CFHA 033	DIVOVF	CCOM 067	LRSTRC	BAL	
326A	C1EF	CFHA 034		036	HTAKE	BR IF S4=1	
326C	A8DE	CFHA 035		CICY 004	CHECK	BR	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
326E	57B9	CFHA 036	HTAKE			T1=D1	
3270	9C70	CFHA 037		BPSW 003	PROG	BR	
		CFHA 038	*				
		CFHA 039	*				
		CFHA 040	*			** LEFT ONE FOR DIVIDE AND HALVE **	
		CFHA 041	*				
		CFHA 042	*				
		CFHA 043	*				
26D8	60DD	CFHA 044	DIVIDL			P1C=P1+P1+C	SHIFT QUOTIENT FOR DIVIDE
26DA	655B	CFHA 045				G1C=G1+G1	SHIFT DIVIDE GUARD
26DC	F467	CFHA 046	LEFT1	051	HSHORT	BR IF G03=1	BR IF SHORT
26DE	611D	CFHA 047				U1C=U1+U1+C	
26E0	600D	CFHA 048				U0C=U0+U0+C	
26E2	633D	CFHA 049				V1C=V1+V1+C	
26E4	622D	CFHA 050				V0C=V0+V0+C	
26E6	699D	CFHA 051	HSHORT			I1C=I1+I1+C	
26E8	688D	CFHA 052				I0C=I0+I0+C	
26EA	6FFD	CFHA 053				H1C=H1+H1+C	
26EC	C1F3	CFHA 054		057	DIV	BR IF S4=1	
26EE	6AAD	CFHA 055				TOC=TO+TO+C	HALVE
26F0	128E	CFHA 056				RTN	ONLY
26F2	6EED	CFHA 057	DIV			H0C=H0+H0+C	DIVIDE
26F4	128E	CFHA 058				RTN	ONLY

 * CROSS REFERENCE FOR CSECT CFHA *

CFHA 006	CFCY 056						
CFHA 014	CFHA 021	CFHA 027					
CFHA 021	CFHA 022						
CFHA 022	CFHA 019						
CFHA 023	CFHA 020						
CFHA 028	CFDV 065	CFMU 061					
CFHA 029	CFAD 058	CFDV 050	CFMD 020	CFMU 053			
CFHA 030	CFHA 023						
CFHA 031	CFHA 012						
CFHA 032	CFAD 121	CFCY 060	CFHA 015	CFHA 028	CFLS 028	CFMU 065	
CFHA 033	CFDV 062	CFMD 015					
CFHA 036	CFHA 034						
CFHA 044	CFDV 022	CFDV 028	CFDV 034				
CFHA 046	CFHA 007	CFHA 008	CFHA 009				
CFHA 051	CFHA 046						
CFHA 057	CFHA 054						

CFLS DESCRIPTIVE TEXT

THE -CFLS- ROUTINE HANDLES THE FOLLOWING OP CODES-

LOAD POSITIVE -20,-30-, (SHORT) AND (LONG)
 LOAD NEGATIVE -21,-31-, (SHORT) AND (LONG)
 LOAD AND TEST -22,-32-, (SHORT) AND (LONG)
 LOAD COMPLEMENT -23,-33-, (SHORT) AND (LONG)
 STORE -60,-70-, (SHORT) AND (LONG)

OBJECTIVES-

LOAD POSITIVE THE SECOND OPERAND IS PLACED IN THE FIRST
 OPERAND LOCATION WITH THE SIGN FORCED
 POSITIVE.

LOAD NEGATIVE THE SECOND OPERAND IS PLACED IN THE FIRST
 OPERAND LOCATION WITH THE SIGN FORCED
 NEGATIVE.

LOAD COMPLEMENT THE SECOND OPERAND IS PLACED IN THE FIRST
 OPERAND LOCATION WITH THE SIGN INVERTED.

LOAD AND TEST THE SECOND OPERAND IS PLACED IN THE FIRST
 OPERAND LOCATION AND THE CONDITION CODE
 IS SET TO INDICATE THE RESULT STATUS.
 CC=0 RESULT FRACTION IS ZERO
 CC=1 RESULT IS LESS THAN ZERO
 CC=2 RESULT IS GREATER THAN ZERO

STORE THE FIRST OPERAND IS STORED AT THE
 SECOND OPERAND LOCATION.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFLS 001	T			LOAD AND STORE OPS	FLOATING POINT
		CFLS 002	*				
		CFLS 003	*				
		CFLS 004	*			*** LOAD POSITIVE **	
		CFLS 005	*				
		CFLS 006	*				
3384	1E85	CFLS 007	LOADP			H0=H0*-K80	MAKE SIGN +
3386	A154	CFLS 008		022	LOADT	BR	
		CFLS 009	*				
		CFLS 010	*				
		CFLS 011	*			*** LOAD NEGATIVE	***
		CFLS 012	*				
		CFLS 013	*				
2150	1E85	CFLS 014	LOADN			H0=H0*-K80	MAKE THE SIGN PLUS
		CFLS 015	*				FALL THROUGH TO COMPLIMENT
		CFLS 016	*				
		CFLS 017	*				
		CFLS 018	*			*** LOAD COMPLIMENT	
		CFLS 019	*				
		CFLS 020	*				
2152	1E8D	CFLS 021	LOADC			H0=H0K80	INVERT SIGN
2154	2C25	CFLS 022	LOADT			P0=0\$K20	SET +
2156	CE5A	CFLS 023		025	OKSOFA	BR IF H00=0	
2158	1C7D	CFLS 024				P0=P0K70	
215A	8BDA	CFLS 025	OKSOFA	CFCO 056	ZTEST	BAL	NOT + CHANGE CC TO - 0101 CC=1
215C	D5E1	CFLS 026		028	CODCOR	BR IF S1=1	GO TEST FRACTION FOR 0
215E	2C07	CFLS 027				P0=0	BR IF FRAC NOT 0 CC IS SET
2160	B266	CFLS 028	CODCOR	CFHA 032	STORE1	BR	SET CC=0 FRAC=0
							GO LOAD R2 TO R1, REST I, GO I CYC

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFLS 029	*				
		CFLS 030	*				
		CFLS 031	*			*** STORE OP ***	
		CFLS 032	*				
		CFLS 033	*				
0D6C	4606	CFLS 034	STORE			D=U	MOVE DESTINATION ADDRESS
0D6E	45BB	CFLS 035				T1=G1H+T1L	R1 OPERAND ADDR
0D70	8212	CFLS 036		CFCO 090	RDOPER	BAL	GET R1 OPER
0D72	7E78	CFLS 037				STH H D+2	
0D74	7878	CFLS 038				STH I D+2	
0D76	F47D	CFLS 039		042	ENDSTO	BR IF G03=1	BA IF SHORT
0D78	7278	CFLS 040				STH V D+2	
0D7A	7070	CFLS 041				STH U D+0	
0D7C	925A	CFLS 042	ENDSTO	CCDM 067	LRSTRC	BAL	RESTORE I
0D7E	A8DE	CFLS 043		CICY 004	CHECK	BR	

 * CROSS REFERENCE FOR CSECT CFLS *

CFLS 007	CFCY 052		
CFLS 014	CFCY 053		
CFLS 021	CFCY 055		
CFLS 022	CFCY 054	CFLS 008	
CFLS 025	CFLS 023		
CFLS 028	CFLS 026		
CFLS 034	CFCY 048		
CFLS 042	CFLS 039		

CFMD DESCRIPTIVE TEXT

THE -CFMD- ROUTINE IS ENTERED FROM THE FLOATING POINT I-CYCLES ROUTINE -CFCY- TO PREPARE FOR THE EXECUTION OF THE FOLLOWING OP CODES -

DIVIDE		MULTIPL
RROP 3D		RROP 3C
RROP 2D		RROP 2C
RXOP 7D		RXOP 7C
RXOP 6D		RXOP 6C

PRIOR TO BRANCHING TO THE PROPER EXECUTE ROUTINE, THE -CFMD- ROUTINE DOES THE FOLLOWING--

1. TEST MULTIPLIER OR DIVISOR FOR ZERO.
2. PRENORMALIZE OPERANDS
3. COMPUTE INTERMEDIATE EXPONENT AND SIGN OF THE RESULT.
4. PREPARE OPERANDS, (FOR FP MULTIPLY) , TO SHARE THE BINARY MULTIPLY COMPUTE LOOP IN ROUTINE -CMLT--.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFMD 001	T			MULTIPLY DIVIDE PRENORMALIZE -FLOATING POINT	
		CFMD 002	*				
		CFMD 003	*				
		CFMD 004	*			2ND OP IS PRENOR. FIRST AND PUT IN A SAVE AREA	
		CFMD 005	*			1ST OP IS THEN PRENOR. AND PUT BACK IN R1 AND	
		CFMD 006	*			RETAINED IN LS H,I,V,U	
		CFMD 007	*				
		CFMD 008	*				
1D7E	3082	CFMD 009	DIVPRE			SET S K=98	SET S0,S3,S4 FOR DIVIDE
1D80	8BDA	CFMD 010	MULPRE	CFCO 056	ZTEST	BAL	CHECK FRACTION FOR 0
1D82	D595	CFMD 011		021	FNOTZ	BR IF S1=1	BR IF PLIER OR DIVISOR IS NOT 0
1D84	D18D	CFMD 012		016	R1ISO	BR IF S5=1	BR IF R1= 0
1D86	F00C	CFMD 013		016	R1ISO	BR IF G07=0	R2 IS 0 BR IF MULT
1D88	27F3	CFMD 014				D1=0\$K0F	IT IS DIVIDE EXCEPTION
1D8A	B268	CFMD 015		CFHA 033	DIVOVF	BR	GO TAKE INT AND STOR 0 ,REST I
1D8C	F011	CFMD 016	R1ISO	019	DONE	BR IF G07=1	BR IF DIVIDE
1D8E	1415	CFMD 017				GO=GO*-K10	MULT TURN OFF SHORT
		CFMD 018	*				FULL 0 STORE FOR MULT.
1D90	5CC2	CFMD 019	DONE			RDH P DA,B8	RESTORE P
1D92	B260	CFMD 020		CFHA 029	DIVNQi	BR	
1D94	D19C	CFMD 021	FNOTZ	025	R2	BR IF S5=0	BR IF USING R2
1D96	5E69	CFMD 022				DO=HO	USING R1 SAVE ORIG EXP
1D98	3E85	CFMD 023				HO=HO\$K80	MAKE EXCESS 192
1D9A	9DA2	CFMD 024		028	NORM	BR	
1D9C	5E79	CFMD 025	R2			D1=HO	SAVE R2 ORIG EXP
1D9E	1E85	CFMD 026				HO=HO*-K80	REMOVE SIGN
1DA0	2E4D	CFMD 027				HO=HO+K40	MAKE EXCESS 128
1DA2	0FF3	CFMD 028	NORM			Z=H1*-K0F	CHECK HI DIGIT FOR 0
1DA4	C4AC	CFMD 029		033	FNORM	BR IF ZNZ	BR IF NORMALIZED
1DA6	A1E4	CFMD 030		CFCO 034	SHIFL4	BAL	
1DA8	2EFF	CFMD 031				HO=HO+KFF	DEC EXP
1DAA	9DA2	CFMD 032		028	NORM	BR	
1DAC	D1BF	CFMD 033	FNORM	042	R1D	BR IF S5=1	
1DAE	3B35	CFMD 034				T1=T1\$K30	DOING R2 SET T TO SAVE AREA ADD
1DB0	5ED9	CFMD 035				P1=HO	SAVE NORM R2 EXP

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1DB2	2EC7	CFMD 036				HO=0	
1DB4	9AD0	CFMD 037		CFCO 078	STORE2	BAL	SAVE NORM R2
1DB6	5AE2	CFMD 038				RDH T DA,BC	T=00X8
1DB8	8212	CFMD 039		CFCO 090	RDCPER	BAL	GET R1 OP
1DBA	2040	CFMD 040				SET S5	
1DBC	5080	CFMD 041		010	MULPRE	BR	
1DBE	5EC9	CFMD 042	R1D			PO=HO	DOING R1,SAVE NORM R1 EXP
1DC0	2EC7	CFMD 043				HO=0	ZERO OUT OPERAND EXPO
1DC2	6761	CFMD 044				D1=D1□D0	COMPUTE RES SIGN D00 BIT
1DC4	17F3	CFMD 045				D1=D1*-K0F	ZERO OUT D1 LOW FOR FLAGS
1DC6	7CDD	CFMD 046				POC=PO%P1+C	COMP IF DIV,TRUE IF MULT
1DC8	57D9	CFMD 047				P1=D1	SAVE RES SIGN
1DCA	F5CF	CFMD 048		050	NOUND	BR IF S3=1	
1DCC	3D13	CFMD 049				P1=P1\$K01	SET UNDERFLOW FLAG
1DCE	7CA2	CFMD 050	NOUND			STH P DA,AC	SAVE RES EXP,SIGN AND FLAGS
1DD0	9AD0	CFMD 051		CFCO 078	STORE2	BAL	PUT NORMALIZED R1 BACK IN R1
1DD2	3B35	CFMD 052				T1=T1\$K30	T=R2 SAVE 0038 OR 0078
1DD4	F058	CFMD 053		055	NDIV	BR IF G07=0	BR IF MULT
1DD6	9F84	CFMD 054		CFDV 006	DIVIDE	BR	
1DD8	10CE	CFMD 055	NDIV			RST S K=FC	CONTINUE MULT PREP.
1DDA	2080	CFMD 056				SET S4	FOR BINARY MULT
1DDC	F46E	CFMD 057		066	LONGM	BR IF G03=0	
1DDE	42E6	CFMD 058				V=H	SAVE
1DE0	4086	CFMD 059				U=I	SHORT CAND R1
1DE2	8212	CFMD 060		CFCO 090	RDCPER	BAL	GET PLIER R2
1DE4	2BC7	CFMD 061				T1=0	SHIFT
1DE6	46E6	CFMD 062				D=H	DATA
1DE8	4E86	CFMD 063				H=I	TO SHARE
1DEA	48A6	CFMD 064				I=T	BINARY
1DEC	8312	CFMD 065		CMLT 027	FLTMU1	BR	BR TO SHARE BINARY MULT
1DEE	4C26	CFMD 066	LONGM			P=V	SAVE
1DF0	4606	CFMD 067				D=U	LOW CAND LONG R1
1DF2	8212	CFMD 068		CFCO 090	RDCPER	BAL	GET LONG PLIER R2
1DF4	4A86	CFMD 069				T=I	SHIFT
1DF6	48E6	CFMD 070				I=H	DATA
1DF8	4E06	CFMD 071				H=U	TO
1DFA	4066	CFMD 072				U=D	SHARE
1DFC	4626	CFMD 073				D=V	BINARY
1DFE	42C6	CFMD 074				V=P	MULT
1E00	8312	CFMD 075		CMLT 027	FLTMU1	BR	BR TO SHARE BINARY MULT

 * CROSS REFERENCE FOR CSECT CFMD *

CFMD 009 CFCY 065
 CFMD 010 CFCY 064 CFMD 041
 CFMD 016 CFMD 012 CFMD 013
 CFMD 019 CFMD 016
 CFMD 021 CFMD 011
 CFMD 025 CFMD 021
 CFMD 028 CFMD 024 CFMD 032
 CFMD 033 CFMD 029
 CFMD 042 CFMD 033
 CFMD 050 CFMD 048

* CROSS REFERENCE FOR CSECT CFMD *

CFMD 055 CFMD 053
CFMD 066 CFMD 057

CFMU DESCRIPTIVE TEXT

FLOATING MULTIPLY OP CODES
 RROP 3C
 RROP 2C
 RXOP 7C
 RXGP 6C

-CFMU- POSTNORMALIZE THE PRODUCT IF NECESSARY. CHECK FOR UNDERFLOW-
 IF UNDERFLOW, SET UNDERFLOW FLAG AND BRANCH TO THE -CFHA- ROUTINE TO SEE IF AN INTERRUPT IS TO BE TAKEN. IF INTERRUPT, ZERO OUT RESULT FRACTION AND EXPONENT AND GO TO THE -BPSW- ROUTINE TO STORE INTERRUPT CODE IN PSW. IF NO INTERRUPT, GO TO THE NORMAL ENTRY TO THE -CFHA- ROUTINE.
 IF NO UNDERFLOW, GO TO THE -CFHA- ROUTINE TO STORE THE NORMALIZED PRODUCT IN THE FIRST OPERAND LOCATION. RETURN TO I-CYCLES ROUTINE -CICY--.

THE FLOATING POINT MULTIPLY ROUTINE, -CFMU- IS ENTERED FROM THE BINARY MULTIPLY ROUTINE -CLMT-. THE FOLLOWING SEQUENCE HAS TAKEN PLACE PRIOR TO ENTRY TO THE -CMPU- ROUTINE--

CFCY- THE MULTIPLY OP CODE IS DECODED AND A BRANCH TO THE -CFMD- ROUTINE IS TAKEN.
 CFMD- THE OPERANDS ARE PRENORMALIZED AND SAVED. THE INTERMEDIATE PRODUCT EXPONENT IS COMPUTED AND SAVED. BRANCH TO THE -CLMT- ROUTINE.
 CLMT- COMPUTE INTERMEDIATE PRODUCT. BRANCH TO THE -CFMU- ROUTINE.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFMU 001	T			FLOATING POINT MULTIPLY	
		CFMU 002	*				
		CFMU 003	*				
		CFMU 004	*				
		CFMU 005	*			THIS ROUTINE IS ONLY REACHED BY A BRANCH FROM BINARY MULTIPLY -IF SHORT MULT IS FINISHED, IF LONG	
		CFMU 006	*			THIS ROUTINE GOES BACK TO BINARY FOR SECOND HALF	
		CFMU 007	*				
		CFMU 008	*				
15F6	5CB2	CFMU 009	MULT			RDH P DA,AE	RESTORE OP CODE TO P
15F8	FC2A	CFMU 010		024	LONG	BR IF P03=0	
15FA	44C6	CFMU 011				G=P ** THIS	IS SHORT MULT IS FIN. PUT OP CODE IN G
15FC	1415	CFMU 012				GO=GO*-K10	MAKE PRODUCT LONG
15FE	5CA2	CFMU 013				RDH P DA,AC	GET RESULT EXPONENT AND FLAGS
1600	5079	CFMU 014				D1=U0	
1602	5FC9	CFMU 015				U0=H1	SHIFT
1604	57F9	CFMU 016				H1=D1	DATA
1606	5189	CFMU 017				I0=U1	FOR
1608	5299	CFMU 018				I1=V0	POSTNORMALIZE
160A	5329	CFMU 019				V0=V1	
160C	5E39	CFMU 020				V1=H0	
160E	21C7	CFMU 021				U1=0	
1610	25C7	CFMU 022				G1=0	ZERO OUT GUARD
1612	95CE	CFMU 023		042	MULTEN	BR	
15AA	5CA2	CFMU 024	LONG			RDH P DA,AC	GET RESULT EXPONENT AND FLAGS
15AC	C93D	CFMU 025		033	LONGEN	BR IF P14=1	BR IF SECOND TIME
15AE	3D83	CFMU 026				P1=P1\$K08	TURN ON SECOND FLAG

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
15B0	7CA2	CFMU 027				STH P DA, AC	PUT FLAGS AND EXPONENT BACK
15B2	5CE2	CFMU 028				RDH P DA, BC	GET R1 ADD
15B4	52C8	CFMU 029				RDH V AS, P+2	GET CAND HI R1
15B6	5EC0	CFMU 030				RDH H AS, P+0	
15B8	2C13	CFMU 031				PO=0\$K01	FOR BINARY MULT
15BA	8356	CFMU 032		CMLT 061	FLTMU2	BR	BR TO SHARE BINARY MULT
15BC	54B2	CFMU 033	LONGEN			RDH G DA, AE	GET OP CODE
15BE	5AF9	CFMU 034				H1=T0	
15C0	5889	CFMU 035				I0=T1	SHIFT
15C2	5699	CFMU 036				I1=D0	DATA
15C4	5359	CFMU 037				G1=V1	FOR
15C6	5039	CFMU 038				V1=U0	POSTNORMALIZE
15C8	5109	CFMU 039				U0=U1	
15CA	5219	CFMU 040				U1=V0	
15CC	5729	CFMU 041				V0=D1	
15CE	46C6	CFMU 042	MULTEN			D=P	SAVE FLAGS
15D0	5CC2	CFMU 043				RDH P DA, B8	GET MASK
15D2	0FF3	CFMU 044				Z=H1*-K0F	CHECK FOR NORMALIZE
15D4	C4DE	CFMU 045		050	ISNORM	BR IF ZNZ	
15D6	A1E4	CFMU 046		CFCO 034	SHIFL4	BAL	GO NORMALIZE
15D8	26FF	CFMU 047				D0=D0+KFF	DEC
15DA	F4DF	CFMU 048		050	ISNORM	BR IF AC=1	
15DC	3713	CFMU 049				D1=D1\$K01	SET UNDERFLOW FLAG
15DE	5AE2	CFMU 050	ISNORM			RDH T DA, BC	GET R1 ADD T=00X8
15E0	8BDA	CFMU 051		CFCO 056	ZTEST	BAL	
15E2	D5E7	CFMU 052		054	RESNOZ	BR IF S1=1	
15E4	B260	CFMU 053		CFHA 029	DIVNOI	BR	STORE TRUE 0 REST 1 GO TO ICYC
15E6	56E9	CFMU 054	RESNOZ			H0=D0	MOVE RESULT EXP
15E8	1E85	CFMU 055				H0=H0*-K80	REMOVE SIGN IN CASE OF DVF
15EA	C76E	CFMU 056		058	PLUS	BR IF D10=0	BR IF RES IS +
15EC	3E85	CFMU 057				H0=H0\$K80	MAKE RES -
15EE	2C80	CFMU 058	PLUS			SET S4	
15F0	F322	CFMU 059		062	NOTUND	BR IF D17=0	
15F2	27D3	CFMU 060				D1=0\$K0D	
15F4	B25E	CFMU 061		CFHA 028	UNDCHK	BR	
15A2	C627	CFMU 062	NOTUND	064	OVF	BR IF D00=1	
15A4	0080	CFMU 063				RST S4	
15A6	27C3	CFMU 064	OVF			D1=0\$K0C	
15A8	B266	CFMU 065		CFHA 032	STORE1	BR	

 * CROSS REFERENCE FOR CSECT CFMU *

CFMU 009	CMLT 063	
CFMU 024	CFMU 010	
CFMU 033	CFMU 025	
CFMU 042	CFMU 023	
CFMU 050	CFMU 045	CFMU 048
CFMU 054	CFMU 052	
CFMU 058	CFMU 056	
CFMU 062	CFMU 059	
CFMU 064	CFMU 062	

THE I-CYCLES ROUTINE IS ENTERED AT FIVE MAJOR ENTRY POINTS.

LABEL CHECKX - ENTRY IS MADE HERE FROM MICROROUTINES THAT NEED CONDITION CODE ZERO SET.

LABEL CHECK - ENTRY HERE IS FROM ROUTINES THAT DID NOT BRANCH OUT TESTING FOR EXECUTE AND INTERRUPT.

LABEL ISTARTO THIS BRANCH SET ENTRY IS MADE FROM ROUTINES THAT TESTED FOR EXECUTE AND INTERRUPT AS THE EXIT FROM THAT ROUTINE WAS MADE.

LABEL ADDERR - ENTRY AT THIS POINT IS MADE WHEN AN ADDRESSING ERROR HAS BEEN DETECTED. THE ERROR IS FLAGGED AND THE *BPSW* ROUTINE IS BRANCHED TO. THE *BPSW* ROUTINE STORES THE OLD PSW, INCLUDING THE INTERRUPT CODE THAT INDICATES AN ADDRESSING ERROR.

LABEL SPECHK - ENTRY IS MADE HERE FROM ROUTINES IN WHICH A SPECIFICATION EXCEPTION IS DETECTED. THE SPECIFICATION EXCEPTION IS FLAGGED AND A BRANCH IS TAKEN TO THE *BPSW* ROUTINE. THE *BPSW* ROUTINE STORES THE OLD PSW, INCLUDING THE INTERRUPTION CODE THAT INDICATES A SPECIFICATION EXCEPTION.

THE ENTRY POINTS CHECKX, CHECK, AND ISTART ARE THE NORMAL ENTRY POINTS TO THE I-CYCLES ROUTINE.

IF THE ENTRY IS MADE FROM AN EXECUTE INSTRUCTION, THE SAVED INSTRUCTION ADDRESS IS RESTORED AND THE NEW INSTRUCTION READ OUT BEGINS.

IF THE ENTRY IS MADE FROM AN EXECUTE INSTRUCTION, AND AN INTERRUPT IS PENDING, THE SAVED INSTRUCTION ADDRESS IS RESTORED AND A BRANCH TO THE *BSWI* ROUTINE IS MADE. THE *BSWI* ROUTINE WILL DETERMINE THE TYPE OF INTERRUPT AND HANDLE IT.

IF THE ENTRY IS MADE WITH ONLY AN INTERRUPT PENDING, THE *BPSW* ROUTINE IS BRANCHED TO DIRECTLY.

IF THE ENTRY IS MADE WITH NO INTERRUPT PENDING AND NO EXECUTE, THE FIRST 2 INSTRUCTION BYTES ARE READ OUT AND A TEST IS MADE ON THE HIGH 4 BITS OF THE OP CODE.

TEST HIGH 4 BITS OF OP CODE

HIGH HEX DIGIT	OPERATION
0,1	TEST LOW 4 BITS OF OP CODE AND BRANCH TO THE PROPER ROUTINE.
2,3,6,7	THESE ARE FLOATING POINT OPS. A BRANCH IS TAKEN DIRECTLY TO THE *CFCY* ROUTINE. THIS IS THE FLOATING POINT I-CYCLES ROUTINE.
4,5	READ 2ND 2 BYTES OF INSTRUCTION, FIND THE EFFECTIVE OPERAND ADDRESS, TEST LOW 4 BITS OF OP CODE, BRANCH TO PROPER ROUTINE.
8	READ 2ND 2 BYTES OF INSTRUCTION, FIND THE EFFECTIVE OPERAND ADDRESS. TEST BIT 4 OF OP CODE, IF A SHIFT OP, BRANCH TO *CSFT*, IF NOT A SHIFT, TEST LOW 4 BITS OF OP CODE AND BRANCH TO PROPER ROUTINE.
9	FIND EFFECTIVE OPERAND ADDRESS, TEST LOW 4 BITS OF OP CODE AND BRANCH TO PROPER ROUTINE.
A,B,C,E	INCREMENT INSTRUCTION ADDRESS +2 FOR A AND B, +4 FOR C AND E. FLAG INVALID OP AND BRANCH TO *BPSW* ROUTINE WHERE THE OPERATION INTERRUPTION CODE IS STORED IN THE OLD PSW.
D	FIND EFFECTIVE OPERAND ADDRESSES, TEST LOW 4 BITS OF OP CODE AND BRANCH TO PROPER ROUTINE.
F	FIND EFFECTIVE OPERAND ADDRESSES, ADD IN THE OPERAND LENGTH CODES TO THE EFFECTIVE ADDRESSES TO POINT TO LOW ORDER BYTES. TEST LOW 4 BITS OF OP CODE AND BRANCH TO PROPER ROUTINE.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CICY 001	T			INSTRUCTION CYCLES 360 OP SET	001
		CICY 002	*				
28DC	2C07	CICY 003	CHECKX			PO=0	SET CC TO EQUAL
28DE	81B3	CICY 004	CHECK	005	ISTART N	N=S BITS67	TEST FOR EXECUTE OR INTERRUPT
29A0	8B7A	CICY 005	ISTART 0	013	EXCHKA	BR	EXECUTE AND INT
29A2	AEF6	CICY 006	ISTART 1	010	EXCHK	BR	EXECUTE NOT INT
29A4	87C8	CICY 007	ISTART 2	BSWI 007	CHECKB	BR	INT NOT EXECUTE
29A6	5498	CICY 008	ISTART 3			RDH G I+2	READ FIRST TWO INST. BYTES
29A8	B461	CICY 009	OPHIBR	016	OPHI N	N=G0H	TEST HIGH OP CODE DIGIT
2EF6	5892	CICY 010	EXCHK			RDH I DA,AA	READ SAVED ADDRESS
2EF8	2020	CICY 011				SET S6	TURN OFF EXECUTE
2EFA	A9A6	CICY 012		008	ISTART 3	BR	
0B7A	5892	CICY 013	EXCHKA			RDH I DA,AA	READ SAVED ADDRESS
0B7C	2020	CICY 014				SET S6	TURN OFF EXECUTE
0B7E	87C8	CICY 015		BSWI 007	CHECKB	BR	GO TO EXCEPTIONAL ROUTINE
2860	B063	CICY 016	OPHI 0	122	OPOX N	N=GOL	RR OP
2862	B043	CICY 017	OPHI 1	147	OP1X N	N=GOL	CODE BREAKDOWN
2864	A844	CICY 018	OPHI 2	CFCY 012	RROP	BR	FLOATING POINT OPS
2866	A844	CICY 019	OPHI 3	CFCY 012	RROP	BR	FLOATING POINT OPS
2868	88DA	CICY 020	OPHI 4	053	OPHI4	BR	RX OP
286A	AE90	CICY 021	OPHI 5	055	OPHI5	BR	CODE BASE CALCULATE
286C	A820	CICY 022	OPHI 6	CFCY 034	RXOP	BR	FLOATING POINT OPS
286E	A820	CICY 023	OPHI 7	CFCY 034	RXOP	BR	FLOATING POINT OPS
2870	9012	CICY 024	OPHI 8	047	OPHI8	BR	RS OPS
2872	8106	CICY 025	OPHI 9	051	OPHI9	BR	SI OPS
2874	A058	CICY 026	OPHI A	065	RSINV	BR	INVALID
2876	A058	CICY 027	OPHI B	065	RSINV	BR	OP
2878	A056	CICY 028	OPHI C	064	SSINV	BR	CODES
287A	88D6	CICY 029	OPHI D	062	OPHID	BR	SS OPS
287C	A056	CICY 030	OPHI E	064	SSINV	BR	INVALID OPS
287E	8AB8	CICY 031	OPHI F	032	SSFX	BR	DECIMAL OPS
0AB8	814C	CICY 032	SSFX	100	BASE3	BAL	
0ABA	55E5	CICY 033				H0=G1XL	CALCULATE
0ABC	61EB	CICY 034				U1C=U1+H0	RIGHT MOST
0ABE	60AD	CICY 035				U0C=U0+T0+C	ADDRESS
0ACO	6FAD	CICY 036				H1C=H1+T0+C	ADD IN LENGTH CODES TO
0AC2	55ED	CICY 037				H0=G1L	POINT TO LOW BYTE
0AC4	63EB	CICY 038				V1C=V1+H0	OF OPERANDS
0AC6	62AD	CICY 039				V0C=V0+T0+C	
0AC8	67AD	CICY 040				D1C=D1+T0+C	
0ACA	6F75	CICY 041				H1=H1\$D1	
0ACC	AB71	CICY 042		044	ADERR N	BR IF H1=NZ	SS FXOP BREAKOUT
0ACE	B01B	CICY 043		241	OPFX N	N=GOL	FLAG ADDRESS ERR
08E0	A9C0	CICY 044	ADERR 0	045	ADDERR 0	BR	MAKE ADDRESSABLE
29C0	2B53	CICY 045	ADDERR 0			T1=0\$K05	FLAG ADDRESS ERROR
29C2	9C70	CICY 046		BPSW 003	PROG	BR	GO TAKE INTERRUPT
1012	5A6A	CICY 047	OPHI8	070	BASE1	BAL	RS OP CODE 8X
1014	C011	CICY 048		050	ALLSHI	BR IF G04=1	SHIFT OPS
1016	9001	CICY 049		195	OP8X N	N=G0 BITS567	8X OPS LOW DIGIT TEST
1010	9BF4	CICY 050	ALLSHI	CSFT 014	SHIFT	BR	SHIFT OPS EXIT
0106	5A6A	CICY 051	OPHI9	070	BASE1	BAL	RS OP CODE 9X
0108	B009	CICY 052		203	OP9X N	N=GOL	BREAKOUT
08DA	AF4E	CICY 053	OPHI4	081	BASE2	BAL	RX GO TO BASE + INDEX

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
08DC	B01D	CICY 054		163	OP4X	N N=GOL	BREAKOUT
2E90	AF4E	CICY 055	OPHI5	081	BASE2	BAL	RX GO TO BASE AND INDEX
2E92	AB53	CICY 056		045	ADDERR	N BR IF H1=NZ	CHECK FOR ADDRESS ERROR
2E94	01C3	CICY 057				Z=U1*-KOC	TEST FOR
2E96	FC8F	CICY 058		061	OPHI5A	BR IF LZ=0	SPECIFICATION CHECK
2E98	2B63	CICY 059	SPECHK			T1=0\$K06	FLAG SPEC CHECK
2E9A	9C70	CICY 060		BPSW 003	PRCG	BR	BRANCH TO STORE INTERRUPT CODE
2E8E	BC71	CICY 061	OPHI5A	179	OP5X	N N=GOL	RX 5XOP BREAKOUT
08D6	814C	CICY 062	OPHID	100	BASE3	BAL	SS DXOP
08D8	B03B	CICY 063		225	OPDX	N N=GOL	BREAKOUT
2056	6884	CICY 064	SSINV			I=I+2	INCREMENT
2058	6884	CICY 065	RSINV			I=I+2	INSTR ADDRESS
205A	2B13	CICY 066	OPERR			T1=0\$K01	AND FLAG ERROR
205C	9C70	CICY 067		BPSW 003	PRCG	BR	BRANCH TO STORE INTERRUPT CODE
335A	2B23	CICY 068	PRIVOP			T1=0\$K02	FLAG PRIV OP
335C	9C70	CICY 069		BPSW 003	PRCG	BR	BRANCH TO STORE INTERRUPT CODE
1A6A	5098	CICY 070	BASE1			RDH U I+2	READ 2ND 2 INST BYTES
1A6C	50BB	CICY 071				T1=UOH	AND PUT IN REG. ADD.
1A6E	C4E7	CICY 072		079	OUT1	BR IF Z=0	BRANCH IF NO BASE
1A70	5EA8	CICY 073				RDH H AS,T+2	GET BASE DATA
1A72	56A0	CICY 074				RDH D AS,T	AND
1A74	617B	CICY 075				U1C=U1+D1	CALCULATE
1A76	606F	CICY 076				UOC=UOL+D0+C	EFFECTIVE
1A78	6FAD	CICY 077				H1C=H1+T0+C	ADDRESS
1A7A	128E	CICY 078				RTN	
1A66	2FC7	CICY 079	OUT1			H1=0	INDICATE ZERO BASE
1A68	128E	CICY 080				RTN	
2F4E	5098	CICY 081	BASE2			RDH U I+2	READ 2ND 2 INST. BYTES
2F50	50BB	CICY 082				T1=UOH	
2F52	C4EF	CICY 083		097	BASE2A	BR IF Z=0	NO BASE BR
2F54	5EA8	CICY 084				RDH H AS,T+2	GET
2F56	56A0	CICY 085				RDH D AS,T	REGISTER DATA
2F58	617B	CICY 086				U1C=U1+D1	CALCULATE
2F5A	606F	CICY 087				UOC=UOL+D0+C	BASE
2F5C	6FAD	CICY 088				H1C=H1+T0+C	VALUE
2F5E	55B3	CICY 089	BASE2B			T1=G1XH	GET INDEX REG.
2F60	C4ED	CICY 090		096	BASE2C	BR IF Z=0	NO INDEX BR
2F62	52A8	CICY 091				RDH V AS,T+2	GET
2F64	56A0	CICY 092				RDH D AS,T	REGISTER DATA
2F66	617B	CICY 093				U1C=U1+D1	CALCULATE
2F68	6C6D	CICY 094				UOC=UO+D0+C	INDEX
2F6A	6F3D	CICY 095				H1C=H1+V1+C	VALUE
2F6C	128E	CICY 096	BASE2C			RTN	
2F6E	2F07	CICY 097	BASE2A			H1=0	INDICATE ZERO BASE
2F70	C4DF	CICY 098		089	BASE2B	BR IF Z=0	UNCONDITIONAL BRANCH
		CICY 099					
014C	5098	CICY 100	BASE3			RDH U I+2	READ 2ND 2 INST. BYTES
014E	50BB	CICY 101				T1=UOH	GET BASE ADDRESS
0150	C4C5	CICY 102		118	BASE3A	BR IF Z=0	BRANCH IF NO BASE 1
0152	5EA8	CICY 103				RDH H AS,T+2	GET
0154	56A0	CICY 104				RDH D AS,T	BASE DATA
0156	617B	CICY 105				U1C=U1+D1	CALCULATE
0158	606F	CICY 106				UOC=UOL+D0+C	1ST

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
015A	6FAD	CICY 107				HIC=H1+T0+C	OPERAND
015C	5298	CICY 108	BASE3B			RDH V 1+2	READ 3RD 2 INST BYTES
015E	528B	CICY 109				T1=VOH	
0160	C4C9	CICY 110		120	BASE3C	BR IF Z=0	BRANCH IF NO BASE 2
0162	6AA4	CICY 111				T=T+2	
0164	56AA	CICY 112				RDH D AS,T-2	
0166	637B	CICY 113				VIC=V1+D1	CALCULATE
0168	626F	CICY 114				VOC=VOL+00+C	2ND
016A	56A0	CICY 115				RDH D AS,T	OPERAND
016C	67AD	CICY 116				DIC=D1+T0+C	
016E	128E	CICY 117				RTN	
0144	2FC7	CICY 118	BASE3A			H1=0	INDICATE ZERO BASE
0146	C4DD	CICY 119		108	BASE3B	BR IF Z=0	UNCONDITIONAL BRANCH
0148	2707	CICY 120	BASE3C			D1=0	INDICATE ZERO BASE
014A	128E	CICY 121				RTN	
2960	A05A	CICY 122	OPOX 0	066	OPERR	BR	*
2962	A05A	CICY 123	OPOX 1	066	OPERR	BR	* INVALID OP CODES
2964	A05A	CICY 124	OPOX 2	066	OPERR	BR	*
2966	AC5A	CICY 125	OPOX 3	066	OPERR	BR	*
2968	AE36	CICY 126	OPOX 4	CSTS 082	RRPO04	BR	SET PROGRAM MASK
296A	A892	CICY 127	OPOX 5	CBRC 023	RRPO05	BR	BRANCH AND LINK RR
296C	ABCE	CICY 128	OPOX 6	CBRC 067	RRPO06	BR	BRANCH ON COUNT RR
296E	A8BA	CICY 129	OPOX 7	CBRC 005	RRPO07	BR	BRANCH CONDITION RR
2970	AB8E	CICY 130	OPOX 8	CSTS 039	RRPO08	BR	SET KEY
2972	A88E	CICY 131	OPOX 9	CSTS 039	RRPO08	BR	INSERT KEY
2974	S8F6	CICY 132	OPOX A	CSTS 093	RRPO0A	BR	SUPERVISOR CALL
2976	AC5A	CICY 133	OPOX B	066	OPERR	BR	*
2978	AC5A	CICY 142	OPOX C	066	OPERR	BR	*
297A	A05A	CICY 143	OPOX D	066	OPERR	BR	* INVALID OP CODES
297C	A05A	CICY 144	OPOX E	066	OPERR	BR	*
297E	A05A	CICY 145	OPOX F	066	OPERR	BR	*
2940	AC34	CICY 147	OP1X 0	CLST 005	RRPO10	BR	LOAD POSITIVE
2942	AC28	CICY 148	OP1X 1	CLST 030	RRPO11	BR	LOAD NEGATIVE
2944	AC0C	CICY 149	OP1X 2	CLST 018	RRPO12	BR	LOAD AND TEST
2946	AC00	CICY 150	OP1X 3	CLST 040	RRPO13	BR	LOAD COMPLEMENT
2948	A91E	CICY 151	OP1X 4	CLOG 069	RRPO14	BR	AND
294A	B36A	CICY 152	OP1X 5	CLOG 036	RRPO15	BR	COMPARE
294C	B372	CICY 153	OP1X 6	CLOG 110	RRPO16	BR	OR
294E	B37A	CICY 154	OP1X 7	CLOG 140	RRPO17	BR	EXCLUSIVE OR
2950	AE9C	CICY 155	OP1X 8	CLST 046	RRPO18	BR	LOAD
2952	9148	CICY 156	OP1X 9	CBIN 005	RRPO19	BR	COMPARE
2954	B366	CICY 157	OP1X A	CBIN 026	RRPO1A	BR	ADD
2956	81F4	CICY 158	OP1X B	CBIN 042	RRPO1B	BR	SUBTRACT
2958	878C	CICY 159	OP1X C	CMLT 007	RRPO1C	BR	MULTIPLY
295A	99F8	CICY 160	OP1X D	CDVD 002	RRPO1D	BR	DIVIDE
295C	B380	CICY 161	OP1X E	CBIN 055	RRPO1E	BR	ADD LOGICAL
295E	B36E	CICY 162	OP1X F	CBIN 076	RRPO1F	BR	SUBTRACT LOGICAL
0E80	AD14	CICY 163	OP4X 0	CLST 055	RXOP40	BR	SUBTRACT HALFWORD
0E82	A996	CICY 164	OP4X 1	CLOG 198	RXOP41	BR	LOAD ADDRESS
0E84	AE88	CICY 165	OP4X 2	CLOG 190	RXOP42	BR	STORE CHARACTER
0E86	AEDC	CICY 166	OP4X 3	CLOG 182	RXOP43	BR	INSERT CHARACTER
0E88	AC56	CICY 167	OP4X 4	CBRC 047	RXOP44	BR	EXECUTE RX
0E8A	A89C	CICY 168	OP4X 5	CBRC 031	RXOP45	BR	BRANCH AND LINK RX

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0E8C	ABD8	CICY 169	OP4X 6	CBRC 073	RXCP46	BR	BRANCH ON COUNT RX
0E8E	A8C2	CICY 170	OP4X 7	CBRC 011	RXCP47	BR	BRANCH ON CONDITION
0E90	AD00	CICY 171	OP4X 8	CLST 066	RXCP48	BR	LOAD HALFWORD
0E92	B388	CICY 172	OP4X 9	CBIN 022	RXCP49	BR	COMPARE HALFWORD
0E94	9A7C	CICY 173	OP4X A	CBIN 038	RXCP4A	BR	ADD HALFWORD
0E96	B176	CICY 174	OP4X B	CBIN 051	RXCP4B	BR	SUBTRACT HALFWORD
0E98	82F8	CICY 175	OP4X C	CMLT 014	RXCP4C	BR	MULTIPLY HALFWORD
0E9A	A05A	CICY 176	OP4X D	066	OPERR	BR	* INVALID OP CODE
0E9C	A9C4	CICY 177	OP4X E	CNVR 005	RXCP4E	BR	CONVERT TO DECIMAL
0E9E	A9C4	CICY 178	OP4X F	CNVR 005	RXCP4E	BR	CONVERT TO BINARY
28E0	AED0	CICY 179	OP5X 0	CLST 078	RXCP50	BR	STORE
28E2	A05A	CICY 180	OP5X 1	066	OPERR	BR	*
28E4	A05A	CICY 181	OP5X 2	066	OPERR	BR	* INVALID OP CODES
28E6	A05A	CICY 182	OP5X 3	066	OPERR	BR	*
28E8	A908	CICY 183	OP5X 4	CLOG 073	RXCP54	BR	AND
28EA	AC9A	CICY 184	OP5X 5	CLOG 039	RXCP55	BR	COMPARE LOGICAL
28EC	B336	CICY 185	OP5X 6	CLOG 114	RXCP56	BR	OR
28EE	B32A	CICY 186	OP5X 7	CLOG 144	RXCP57	BR	EXCLUSIVE OR
28F0	AEC4	CICY 187	OP5X 8	CLST 086	RXCP58	BR	LOAD
28F2	A8C8	CICY 188	OP5X 9	CBIN 008	RXCP59	BR	COMPARE
28F4	AC18	CICY 189	OP5X A	CBIN 030	RXCP5A	BR	ADD
28F6	810A	CICY 190	OP5X B	CBIN 046	RXCP5B	BR	SUBTRACT
28F8	8784	CICY 191	OP5X C	CMLT 002	RXCP5C	BR	MULTIPLY
28FA	A218	CICY 192	OP5X D	CDVD 006	RXCP5D	BR	DIVIDE
28FC	ADC6	CICY 193	OP5X E	CBIN 059	RXCP5E	BR	ADD LOGICAL
28FE	AF72	CICY 194	OP5X F	CBIN 080	RXCP5F	BR	SUBTRACT LOGICAL
1000	ABC2	CICY 195	OP8X 0	CSTS 011	RSOP80	BR	SET SYSTEM MASK
1002	A05A	CICY 196	OP8X 1	066	OPERR	BR	* INVALID OP CODE
1004	AB7E	CICY 197	OP8X 2	CSTS 003	RSOP82	BR	LOAD PSW
1006	AB8A	CICY 198	OP8X 3	CSTS 019	RSOP83	BR	DIAGNOSE
1008	B302	CICY 199	OP8X 4	CSTS 104	RSOP84	BR	WRITE DIRECT
100A	B30C	CICY 200	OP8X 5	CSTS 109	RSOP85	BR	READ DIRECT
100C	ACC8	CICY 201	OP8X 6	CBRC 090	RSOP86	BR	BRANCH ON INDEX HIGH
100E	ACC8	CICY 202	OP8X 7	CBRC 090	RSOP86	BR	BRANCH ON INDEX LOW OR EQUAL
0400	AE68	CICY 203	OP9X 0	CLST 094	RSOP90	BR	STORE MULTIPLE
0402	ACB0	CICY 204	OP9X 1	CLOG 170	RSOP91	BR	TEST UNDER MASK
0404	A93A	CICY 205	OP9X 2	CLOG 004	RSOP92	BR	MOVE
0406	AE6C	CICY 206	OP9X 3	CSTS 071	RSOP93	BR	TEST AND SET
0408	AE22	CICY 207	OP9X 4	CLOG 086	RSCP94	BR	AND
040A	ACTA	CICY 208	OP9X 5	CLOG 045	RSOP95	BR	COMPARE LOGICAL
040C	ABF6	CICY 209	OP9X 6	CLOG 122	RSOP96	BR	OR
040E	AD72	CICY 210	OP9X 7	CLOG 152	RSOP97	BR	EXCLUSIVE OR
0410	9AB0	CICY 211	OP9X 8	CLST 103	RSOP98	BR	LOAD MULTIPLE
0412	A05A	CICY 217	OP9X 9	066	OPERR	BR	*
0414	A05A	CICY 219	OP9X A	066	OPERR	BR	* INVALID OP CODES
0416	A05A	CICY 220	OP9X B	066	OPERR	BR	*
0418	8CB8	CICY 221	OP9X C	DCLA 002	IOINST	BR	START I/O
041A	8CB8	CICY 222	OP9X D	DCLA 002	IOINST	BR	TEST I/O
041C	8CB8	CICY 223	OP9X E	DCLA 002	IOINST	BR	HALT I/O
041E	8CB8	CICY 224	OP9X F	DCLA 002	IOINST	BR	TEST CHANNEL
0DA0	A05A	CICY 225	OPDX 0	066	OPERR	BR	* INVALID OP CODE
0DA2	AE48	CICY 226	OPDX 1	CLOG 016	SSOPD1	BR	MOVE NUMERIC
0DA4	AEAA	CICY 227	OPDX 2	CLOG 008	SSOPD2	BR	MOVE

CLOG -- LOGICAL OPERATIONS

* ENTRY	OPERATION	BYTE 1	BYTE 2	BYTE 3	BYTE 4
* RROP14	AND	14	R1 R2	.	.
* RROP15	COMPARE	15	R1 R2	.	.
* RROP16	OR	16	R1 R2	.	.
* RROP17	EXCLUSIVE OR	17	R1 R2	.	.
* RXOP41	LOAD ADDRESS	41	R1 X2	B2 D2	D2
* RXOP42	STORE CHARACTER	42	R1 X2	B2 D2	D2
* KXOP43	INSERT CHARACTER	43	R1 X2	B2 D2	D2
* RXOP54	AND	54	R1 X2	B2 D2	D2
* RXOP55	COMPARE LOGICAL	55	R1 X2	B2 D2	D2
* RXOP56	OR	56	R1 X2	B2 D2	D2
* RXOP57	EXCLUSIVE OR	57	R1 X2	B2 D2	D2
* RSOP91	TEST UNDER MASK	91	I2	B1 D1	D1
* RSOP92	MOVE	92	I2	B1 D1	D1
* RSOP94	AND	94	I2	B1 D1	D1
* RSOP95	COMPARE LOGICAL	95	I2	B1 D1	D1
* RSOP96	OR	96	I2	B1 D1	D1
* RSOP97	EXCLUSIVE OR	97	I2	B1 D1	D1

OBJECTIVES

- AND, OR, EXCLUSIVE OR
THE LOGIC FUNCTION IS PERFORMED WITH CORRESPONDING BITS IN THE 1ST AND 2ND OPERANDS. THE RESULT REPLACES THE 1ST OPERAND. IN THE SI FORMAT, I2 IS THE 2ND OPERAND AND IS ONE BYTE LONG.
- COMPARE
COMPARE 1ST OPERAND TO THE 2ND OPERAND. OPERANDS ARE TREATED AS UNSIGNED BINARY QUANTITIES.
- STORE CHARACTER
MOVE BYTE 3 OF R1 TO PROGRAM STORAGE ADDRESSED BY 2ND OPERAND.
- MOVE
THE 2ND OPERAND IS PLACED IN THE 1ST OPERAND LOCATION. IN MOVE IMMEDIATE, I2 IS THE 2ND OPERAND.
- MOVE NUMERIC
SAME AS MOVE, EXCEPT ONLY THE LOW-ORDER 4 BITS ARE MOVED.
- MOVE ZONES
SAME AS MOVE, EXCEPT ONLY THE HIGH-ORDER 4 BITS ARE MOVED.
- TEST UNDER MASK
THE IMMEDIATE DATA IN BYTE 2 OF INSTRUCTION IS TESTED BIT FOR BIT AGAINST THE 1ST OPERAND.
- LOAD ADDRESS
SET THE EFFECTIVE ADDRESS OF THE 2ND OPERAND INTO THE LOW 24 BITS OF R1.

SS FORMAT INSTRUCTIONS

* OP CODE	L1	L2	B1	D1	D1	D1	B2	D2	D2	D2
SSOPD1										
SSOPD2										
SSOPD3										
SSOPD4										
SSOPD5										
SSOPD6										
SSOPD7										

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CLOG 001	T			LOGICAL OPERATIONS	
		CLOG 002	*				
		CLOG 003	*			MOVE IMMEDIATE	
293A	AB53	CLOG 004	RSOP92	CICY 045	ADDERR N	BR IF H1=NZ	MOVE
293C	7510	CLOG 005				STB G1 U	IMMEDIATE
293E	81B3	CLOG 006		CICY 005	ISTART N	N=S BITS67	EXECUTION
		CLOG 007	*			MOVE CHARACTERS	
2EAA	6F75	CLOG 008	SSOPD2			H1=H1&D1	MOVE
2EAC	AB53	CLOG 009		CICY 045	ADDERR N	BR IF H1=NZ	CHARACTERS

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2EAE	5738	CLOG 010	OPD2			RDB D1 V+1	EXECUTION
2E80	7718	CLOG 011				STB D1 U+1	
2EB2	25FF	CLOG 012				G1=G1+KFF	
2EB4	F4AF	CLOG 013		010	OPD2	BR IF AC=1	
2EB6	81B3	CLOG 014		CICY 005	ISTART N	N=S BITS67	
		CLOG 015	*			MOVE NUMERICS	
2E48	6F75	CLOG 016	SSOPD1			H1=H1\$D1	MOVE
2E4A	AB53	CLOG 017		CICY 045	ADDERR N	BR IF H1=NZ	NUMERICS
2E4C	5738	CLOG 018	OPD1			RDB D1 V+1	EXECUTION
2E4E	5F10	CLOG 019				RDB H1 U	
2E50	4F7B	CLOG 020				D1=H1H+D1L	
2E52	7718	CLOG 021				STB D1 U+1	
2E54	25FF	CLOG 022				G1=G1+KFF	
2E56	F4CD	CLOG 023		018	OPD1	BR IF AC=1	
2E58	81B3	CLOG 024		CICY 005	ISTART N	N=S BITS67	
		CLOG 025	*			MOVE ZONES	
2E5A	6F75	CLOG 026	SSOPD3			H1=H1\$D1	MOVE
2E5C	AB53	CLOG 027		CICY 045	ADDERR N	BR IF H1=NZ	ZONES
2E5E	5738	CLOG 028	OPD3			RDB D1 V+1	EXECUTION
2E60	5F10	CLOG 029				RDB H1 U	
2E62	4F7D	CLOG 030				D1=H1L+D1H	
2E64	7718	CLOG 031				STB D1 U+1	
2E66	25FF	CLOG 032				G1=G1+KFF	
2E68	F4DF	CLOG 033		028	OPD3	BR IF AC=1	
2E6A	81B3	CLOG 034		CICY 005	ISTART N	N=S BITS67	
		CLOG 035	*			COMPARE LOGICAL RR	COMPARE
336A	8A70	CLOG 036	RROP15	CCOM 036	CRGET	BAL	LOGICAL
336C	AC9C	CLOG 037		040	CXCAML	BR	INSTRUCTIONS
		CLOG 038	*			COMPARE LOGICAL RX	EXECUTION
2C9A	A770	CLOG 039	RXOP55	CCOM 045	CXGET	BAL	
2C9C	8174	CLOG 040	CXCAML	CCOM 086	CSLOOP	BAL	
2C9E	D5A8	CLOG 041		054	OP95B	BR IF S1=0	GO SET CC0
2CA0	F5A5	CLOG 042		052	OP95A	BR IF S3=1	GO SET CC2
2CA2	AC84	CLOG 043		050	OP95C	BR	GO SET CC1
		CLOG 044	*			COMPARE LOGICAL RS	
2C7A	AB53	CLOG 045	RSOP95	CICY 045	ADDERR N	BR IF H1=NZ	
2C7C	5F10	CLOG 046				RDB H1 U	
2C7E	7F51	CLOG 047				H1=H1-G1+1	
2C80	C4A9	CLOG 048		054	OP95B	BR IF Z=0	
2C82	F4A5	CLOG 049	OP95D	052	OP95A	BR IF AC=1	
2C84	2C55	CLOG 050	OP95C			PO=0\$K50	
2C86	81B3	CLOG 051		CICY 005	ISTART N	N=S BITS67	
2CA4	2C25	CLOG 052	OP95A			PO=0\$K20	
2CA6	81B3	CLOG 053		CICY 005	ISTART N	N=S BITS67	
2CA8	2C07	CLOG 054	OP95B			PO=0	
2CAA	81B3	CLOG 055		CICY 005	ISTART N	N=S BITS67	
		CLOG 056	*				
		CLOG 057	*			COMPARE LOGICAL CHARACTERS	
2C88	6F75	CLOG 058	SSOPD5			H1=H1\$D1	
2C8A	AB53	CLOG 059		CICY 045	ADDERR N	BR IF H1=NZ	
2C8C	5738	CLOG 060	OPD5			RDB D1 V+1	
2C8E	5F18	CLOG 061				RDB H1 U+1	
2C90	7F71	CLOG 062				H1=H1-D1+1	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT
2C92	C482	CLOG 063		049	OP95D	BR IF ZNZ
2C94	25FF	CLOG 064				G1=G1+KFF
2C96	F48D	CLOG 065		060	OPD5	BR IF AC=1
2C98	ACA8	CLOG 066		054	OP95B	BR
		CLOG 067	*			
		CLOG 068	*	AND	RR	
291E	8A70	CLOG 069	RROP14	CCOM 036	CRGET	BAL
2920	A9CA	CLOG 070		074	CXAND	BR
		CLOG 071	*			
		CLOG 072	*	AND	RX	
2908	A770	CLOG 073	RXOP54	CCOM 045	CXGET	BAL
290A	6F17	CLOG 074	CXAND			H1=H1*U1
290C	6EC7	CLOG 075				H0=H0*U0
290E	6737	CLOG 076				D1=D1*V1
2910	6627	CLOG 077				DO=DO*V0
2912	76A8	CLOG 078	CCLOST			STH D AS,T+2
2914	7EAA	CLOG 079				STH H AS,T-2
2916	E59B	CLOG 080	S2CHK	082	SETNZ	BR IF S2=1
2918	ACA8	CLOG 081		054	OP95B	BR
291A	2C55	CLOG 082	SETNZ			PO=0\$K50
291C	81B3	CLOG 083		CICY 005	ISTART N	N=S BITS67
		CLOG 084	*			
		CLOG 085	*	AND	RS	
2E22	AB53	CLOG 086	RSOP94	CICY 045	ADDERR N	BR IF H1=NZ
2E24	5F10	CLOG 087				RDB H1 U
2E26	6F57	CLOG 088				H1=H1*G1
2E28	C4B1	CLOG 089	OP94A	093	CP94B	BR IF Z=0
2E2A	7F10	CLOG 090				STB H1 U
2E2C	2C55	CLOG 091				PO=0\$K50
2E2E	81B3	CLOG 092		CICY 005	ISTART N	N=S BITS67
2E30	7F10	CLOG 093	OP94B			STB H1 U
2E32	2C07	CLOG 094				PO=0
2E34	81B3	CLOG 095		CICY 005	ISTART N	N=S BITS67
		CLOG 096	*			
		CLOG 097	*	AND	SS	
29AA	6F75	CLOG 098	SSOPD4			H1=H1\$D1
29AC	AB53	CLOG 099		CICY 045	ADDERR N	BR IF H1=NZ
29AE	10CE	CLOG 100				RST S K=FC
29B0	5738	CLOG 101	OPD4			RDB D1 V+1
29B2	5F10	CLOG 102				RDB H1 U
29B4	67F7	CLOG 103				D1=D1*H1
29B6	7718	CLOG 104				STB D1 U+1
29B8	25FF	CLOG 105				G1=G1+KFF
29BA	F4B1	CLOG 106		101	OPD4	BR IF AC=1
29BC	A916	CLOG 107		080	S2CHK	BR
		CLOG 108	*			
		CLOG 109	*	OR	RR	
3372	8A70	CLOG 110	RROP16	CCOM 036	CRGET	BAL
3374	B338	CLOG 111		115	CXOR	BR
		CLOG 112	*			
		CLOG 113	*	OR	RX	
3336	A770	CLOG 114	RXOP56	CCOM 045	CXGET	BAL
3338	6F15	CLOG 115	CXOR			H1=H1\$U1

AND
EXECUTION

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
333A	6E05	CLOG 116				H0=H0\$U0	
333C	6735	CLOG 117				D1=D1\$V1	
333E	6625	CLOG 118				D0=D0\$V0	
3340	A912	CLOG 119		078	CCLOST	BR	
		CLOG 120	*				
		CLOG 121	*				
2BF6	AB53	CLOG 122	RSOP96	OR	RS		
2BF8	5F10	CLOG 123		CICY 045	ADDERR N	BR IF H1=NZ	
2BFA	6F55	CLOG 124				RDB H1 U	
2BFC	AE28	CLOG 125		089	OP94A	H1=H1\$G1	
		CLOG 126	*			BR	
		CLOG 127	*				
				OR	SS		
2DFA	6F75	CLOG 128	SSOPD6			H1=H1\$D1	
2DFC	AB53	CLOG 129		CICY 045	ADDERR N	BR IF H1=NZ	
2DFE	10CE	CLOG 130				RST S K=FC	
2E00	5738	CLOG 131	OPD6			RDB D1 V+1	
2E02	5F10	CLOG 132				RDB H1 U	
2E04	67F5	CLOG 133				D1=D1\$H1	
2E06	7718	CLOG 134				STB D1 U+1	
2E08	25FF	CLOG 135				G1=G1+KFF	
2E0A	F481	CLOG 136		131	OPD6	BR IF AC=1	
2E0C	A916	CLOG 137		080	S2CHK	BR	
		CLOG 138	*				
		CLOG 139	*				
				XOR	RR		
337A	8A70	CLOG 140	RROP17	CCOM 036	CRGET	BAL	XOR
337C	B32C	CLOG 141		145	CXXOR	BR	EXECUTION
		CLOG 142	*				
		CLOG 143	*				
				XOR	RX		
332A	A770	CLOG 144	RXOP57	CCOM 045	CXGET	BAL	
332C	6F11	CLOG 145	CXXOR			H1=H1\$U1	
332E	6E01	CLOG 146				H0=H0\$U0	
3330	6731	CLOG 147				D1=D1\$V1	
3332	6621	CLOG 148				D0=D0\$V0	
3334	A912	CLOG 149		078	CCLOST	BR	
		CLOG 150	*				
		CLOG 151	*				
				XOR	RS		
2D72	AB53	CLOG 152	RSOP97	CICY 045	ADDERR N	BR IF H1=NZ	
2D74	5F10	CLOG 153				RDB H1 U	
2D76	6F51	CLOG 154				H1=H1\$G1	
2D78	AE28	CLOG 155		089	OP94A	BR	
		CLOG 156	*				
		CLOG 157	*				
				XOR	SS		
2DE0	6F75	CLOG 158	SSOPD7			H1=H1\$D1	
2DE2	AB53	CLOG 159		CICY 045	ADDERR N	BR IF H1=NZ	
2DE4	10CE	CLOG 160				RST S K=FC	
2DE6	5738	CLOG 161	OPD7			RDB D1 V+1	
2DE8	5F10	CLOG 162				RDB H1 U	
2DEA	67F1	CLOG 163				D1=D1\$H1	
2DEC	7718	CLOG 164				STB D1 U+1	
2DEE	25FF	CLOG 165				G1=G1+KFF	
2DF0	F4E7	CLOG 166		161	OPD7	BR IF AC=1	
2DF2	A916	CLOG 167		080	S2CHK	BR	
		CLOG 168	*				

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CLOG 169	*			TEST UNDER MASK	
2CB0	AB53	CLOG 170	RSOP91	CICY 045	ADDERR N	BR IF H1=NZ	
2CB2	5F10	CLOG 171				RDB H1 U	
2CB4	6F57	CLOG 172				H1=H1*G1	
2CB6	C4A9	CLOG 173		054	OP95B	BR IF Z=0	
2CB8	6F51	CLOG 174				H1=H1*G1	
2CBA	C4AD	CLOG 175		178	OP91B	BR IF Z=0	
2CBC	2C55	CLOG 176				PO=0\$K50	
2CBE	81B3	CLOG 177		CICY 005	ISTART N	N=S BITS67	
2CAC	2C75	CLOG 178	OP91B			PO=0\$K70	
2CAE	81B3	CLOG 179		CICY 005	ISTART N	N=S BITS67	
		CLOG 180	*				
		CLOG 181	*			INSERT CHARACTER	
2EDC	AB53	CLOG 182	RXOP43	CICY 045	ADDERR N	BR IF H1=NZ	
2EDE	55BB	CLOG 183				T1=G1H	
2EE0	2B38	CLOG 184				T1=T1+K03	
2EE2	5710	CLOG 185				RDB D1 U	
2EE4	77A0	CLOG 186				STB D1 AS,T	
2EE6	81B3	CLOG 187		CICY 005	ISTART N	N=S BITS67	
		CLOG 188	*				
		CLOG 189	*			STORE CHARACTER	
2EB8	AB53	CLOG 190	RXOP42	CICY 045	ADDERR N	BR IF H1=NZ	
2EBA	55BB	CLOG 191				T1=G1H	
2EBC	2B38	CLOG 192				T1=T1+K03	
2EBE	57AC	CLOG 193				RDB D1 AS,T	
2ECO	7710	CLOG 194				STB D1 U	
2EC2	81B3	CLOG 195		CICY 005	ISTART N	N=S BITS67	
		CLOG 196	*				
		CLOG 197	*			LOAD ADDRESS	
2996	55BB	CLOG 198	RXOP41			T1=G1H	
2998	2EC7	CLOG 199				H0=0	
299A	7EA8	CLOG 200				STH H AS,T+2	
299C	70A0	CLOG 201				STH U AS,T	
299E	81B3	CLOG 202		CICY 005	ISTART N	N=S BITS67	

 * CROSS REFERENCE FOR CSECT CLOG *

CLOG 004	CICY 205						
CLOG 008	CICY 227						
CLOG 010	CLOG 013						
CLOG 016	CICY 226						
CLOG 018	CLOG 023						
CLOG 026	CICY 228						
CLOG 028	CLOG 033						
CLOG 036	CICY 152						
CLOG 039	CICY 184						
CLOG 040	CLOG 037						
CLOG 045	CICY 208						
CLOG 049	CLOG 063						
CLOG 050	CLOG 043	CSAS 103	CTRT 032	CTRT 062			
CLOG 052	CFAD 078	CLOG 042	CLOG 049	CSAS 102	CSFT 085	CTRT 031	CTRT 063
CLOG 054	CLOG 041	CLOG 048	CLOG 066	CLOG 081	CLOG 173		
CLOG 058	CICY 230						

* CROSS REFERENCE FOR CSECT CLOG *

CLOG 060	CLOG 065		
CLOG 069	CICY 151		
CLOG 073	CICY 183		
CLOG 074	CLOG 070		
CLOG 078	CLOG 119	CLOG 149	
CLOG 080	CLOG 107	CLOG 137	CLOG 167
CLOG 082	CFAD 079	CLOG 080	CSFT 076
CLOG 086	CICY 207		
CLOG 089	CLOG 125	CLOG 155	
CLOG 093	CLOG 089		
CLOG 098	CICY 229		
CLOG 101	CLOG 106		
CLOG 110	CICY 153		
CLOG 114	CICY 185		
CLOG 115	CLOG 111		
CLOG 122	CICY 209		
CLOG 128	CICY 231		
CLOG 131	CLOG 136		
CLOG 140	CICY 154		
CLOG 144	CICY 186		
CLOG 145	CLOG 141		
CLOG 152	CICY 210		
CLOG 158	CICY 232		
CLOG 161	CLOG 166		
CLOG 170	CICY 204		
CLOG 178	CLOG 175		
CLOG 182	CICY 166		
CLOG 190	CICY 165		
CLOG 198	CICY 164		

CLST -- LOAD AND STORE INSTRUCTIONS

* INSTRUCTION FORMAT *						
* ENTRY	* OPERATION	* BYTE 1	* BYTE 2	* BYTE 3	* BYTE 4	* *
* RROP10	. LOAD POSITIVE	. 10	. R1 R2	.	.	* *
* RROP11	. LOAD NEGATIVE	. 11	. R1 R2	.	.	* *
* RROP12	. LOAD AND TEST	. 12	. R1 R2	.	.	* *
* RROP13	. LOAD COMPLEMENT	. 13	. R1 R2	.	.	* *
* RROP18	. LOAD	. 18	. R1 R2	.	.	* *
* RSOP90	. STORE MULTIPLE	. 90	. R1 R3	. B2 D2	. D2	* *
* RSOP98	. LOAD MULTIPLE	. 98	. R1 R3	. B2 D2	. D2	* *
* RXOP40	. STORE HALFWORD	. 40	. R1 X2	. B2 D2	. D2	* *
* RXOP48	. LOAD HALFWORD	. 48	. R1 X2	. B2 D2	. D2	* *
* RXOP50	. STORE	. 50	. R1 X2	. B2 D2	. D2	* *
* RXOP58	. LOAD	. 58	. R1 X2	. B2 D2	. D2	* *

OBJECTIVES CONTINUED

- LOAD AND TEST - THE 2ND OPERAND IS PLACED IN THE 1ST OPERAND LOCATION, AND THE SIGN AND THE MAGNITUDE OF THE 2ND OPERAND DETERMINE THE CONDITION CODE.
- LOAD HALFWORD - PLACE THE HALFWORD 2ND OPERAND INTO THE 1ST OPERAND LOCATION. EXTEND THE SIGN INTO THE HIGH-ORDER 16 BITS OF THE 1ST OPERAND.
- LOAD MULTIPLE - THE INFORMATION AT THE EFFECTIVE 2ND OPERAND ADDRESS IS LOADED INTO GENERAL REGISTER STARTING WITH THE ONE SPECIFIED BY R1 AND ENDING WITH THE ONE SPECIFIED BY R3.
- STORE MULTIPLE - THE INFORMATION IN THE GENERAL REGISTER STARTING WITH THE REGISTER SPECIFIED BY R1 AND ENDING WITH THE REGISTER SPECIFIED BY R3 IS STORED AT THE LOCATIONS DESIGNATED BY THE EFFECTIVE 2ND OPERAND ADDRESS.
- STORE HALFWORD - MOVE THE 16 LOW-ORDER BYTES OF THE 1ST OPERAND TO PROGRAM STORAGE ADDRESSED BY THE 2ND OPERAND.
- STORE - THE 1ST OPERAND IS MOVED TO PROGRAM STORAGE AT THE EFFECTIVE ADDRESS OF THE 2ND OPERAND.

OBJECTIVES

- LOAD POSITIVE -- THE TRUE VALUE OF THE 2ND OPERAND IS PLACED IN THE 1ST OPERAND LOCATION.
- LOAD NEGATIVE - THE TWOS COMPLIMENT OF THE TRUE VALUE OF THE 2ND OPERAND IS PLACED IN THE 1ST OPERAND LOCATION.
- LOAD COMPLEMENT - THE TWOS COMPLIMENT OF THE 2ND OPERAND IS PLACED IN THE 1ST OPERAND LOCATION.
- LOAD - PLACE THE 2ND OPERAND INTO THE 1ST OPERAND LOCATION.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CLST 001	T			LOAD AND STORE INSTRUCTIONS	
		CLST 002	*				
		CLST 003	*				
		CLST 004	*			LOAD POSITIVE	
2C34	837C	CLST 005	RROP10	CCOM 029	LGET	BAL	
2C36	C610	CLST 006		020	DCHECK	BR IF D00=0	
2C38	8AE0	CLST 007	PCOMPL	CCOM 076	COMPL	BAL	
2C3A	C610	CLST 008		020	DCHECK	BR IF D00=0	
2C3C	76A8	CLST 009	OVCHEK			STH D AS,T+2	
2C3E	7EAA	CLST 010				STH H AS,T-2	
2C40	2C75	CLST 011	SHFTOV			PO=0\$K70	SET CC=3-OVERFLOW
2C42	CDG9	CLST 012		014	TAKEOV	BR IF P10=1	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT
2C44	81B3	CLST 013		CICY 005	ISTART N	N=S BITS67
2C08	2B83	CLST 014	TAKEOV			T1=0\$K08
2C0A	9C70	CLST 015		BPSW 003	PROG	BR
		CLST 016	*			
		CLST 017	*		LOAD AND TEST	
2C0C	837C	CLST 018	RROP12	CCOM 029	LGET	BAL
2C0E	C625	CLST 019	DBRAN	036	NEGCC1	BR IF D00=1
2C10	4EE6	CLST 020	DCHECK			H=H
2C12	4666	CLST 021				D=D
2C14	D5C7	CLST 022		024	TESCC2	BR IF S1=1
2C16	AC30	CLST 023		034	NEGCC0	BR
2C46	2C25	CLST 024	TESCC2			PO=0\$K20
2C48	76A8	CLST 025	DSTORE			STH D AS,T+2
2C4A	7EAA	CLST 026				STH H AS,T-2
2C4C	81B3	CLST 027		CICY 005	ISTART N	N=S BITS67
		CLST 028	*			
		CLST 029	*		LOAD NEGATIVE	
2C28	837C	CLST 030	RROP11	CCOM 029	LGET	BAL
2C2A	C625	CLST 031		036	NEGCC1	BR IF D00=1
2C2C	8AE0	CLST 032		CCOM 076	COMPL	BAL
2C2E	C625	CLST 033		036	NEGCC1	BR IF D00=1
2C30	2C07	CLST 034	NEGCC0			PO=0
2C32	AC48	CLST 035		025	DSTORE	BR
2C24	2C55	CLST 036	NEGCC1			PO=0\$K50
2C26	AC48	CLST 037		025	DSTORE	BR
		CLST 038	*			
		CLST 039	*		LOAD COMPLIMENT	
2C00	837C	CLST 040	RROP13	CCOM 029	LGET	BAL
2C02	C639	CLST 041		007	PCCMPL	BR IF D00=1
2C04	8AE0	CLST 042		CCOM 076	COMPL	BAL
2C06	AC0E	CLST 043		019	DBRAN	BR
		CLST 044	*			
		CLST 045	*		LOAD OP RR	
2E9C	55B3	CLST 046	RROP18			T1=G1XH
2E9E	56A8	CLST 047				RDH D AS,T+2
2EA0	5EAA	CLST 048				RDH H AS,T-2
2EA2	55BB	CLST 049				T1=G1H
2EA4	76A8	CLST 050				STH D AS,T+2
2EA6	7EAA	CLST 051				STH H AS,T-2
2EA8	81B3	CLST 052		CICY 005	ISTART N	N=S BITS67
		CLST 053	*			
		CLST 054	*		STORE HALFWORD	
2D14	AB53	CLST 055	RXOP40	CICY 045	ADDERR N	BR IF H1=NZ
2D16	01E3	CLST 056				Z=U1*-K0E
2D18	FCA4	CLST 057		063	SPEC	BR IF LZNZ
2D1A	55BB	CLST 058				T1=G1H
2D1C	2B2B	CLST 059				T1=T1+K02
2D1E	52A0	CLST 060				RDH V AS,T+0
2D20	7210	CLST 061				STH V U+0
2D22	81B3	CLST 062	RSEND	CICY 005	ISTART N	N=S BITS67
2D24	AE98	CLST 063	SPEC	CICY 059	SPECHK	BR
		CLST 064	*			
		CLST 065	*		LOAD HALFWORD	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT
2D00	AB53	CLST 066	RXOP48	CICY 045	ADDERR N	BR IF H1=NZ
2D02	01E3	CLST 067				Z=U1*-KOE
2D04	FOA4	CLST 068		063	SPEC	BR IF LZNZ
2D06	55BB	CLST 069				T1=G1H
2J08	5E10	CLST 070				RDH H U
2D0A	2707	CLST 071				D1=0
2D0C	CE10	CLST 072		074	EXT	BR IF H00=0
2D0E	27F7	CLST 073				D1=0\$KFF
2D10	5769	CLST 074	EXT			D0=D1
2D12	AC48	CLST 075		025	DSTORE	BR
		CLST 076	*			
		CLST 077	*		STORE OP RX	
2ED0	55BB	CLST 078	RXOP50			T1=G1H
2ED2	56A8	CLST 079				RDH D AS,T+2
2ED4	5EAA	CLST 080				RDH H AS,T-2
2ED6	7618	CLST 081				STH D U+2
2ED8	7E10	CLST 082				STH H U
2EDA	81B3	CLST 083		CICY 005	ISTART N	N=S BITS67
		CLST 084	*			
		CLST 085	*		LOAD OP RX	
2EC4	55BB	CLST 086	RXOP58			T1=G1H
2EC6	5618	CLST 087				RDH D U+2
2EC8	5E10	CLST 088				RDH H U
2ECA	76A8	CLST 089				STH D AS,T+2
2ECC	7EA0	CLST 090				STH H AS,T
2ECE	81B3	CLST 091		CICY 005	ISTART N	N=S BITS67
		CLST 092	*			
		CLST 093	*		LOAD MULTIPLE	
2EE8	AD26	CLST 094	RSOP90	110	RS9XA	BAL
2EEA	52A8	CLST 095	RS90A			RDH V AS,T+2
2EEC	7218	CLST 096				STH V U+2
2EEE	52A0	CLST 097				RDH V AS,T
2EFO	7218	CLST 098				STH V U+2
2EF2	AD30	CLST 099		115	RS9XB	BAL
2EF4	AEEA	CLST 100		095	RS90A	BR
		CLST 101	*			
		CLST 102	*		STORE MULTIPLE	
1AB0	AD26	CLST 103	RSOP98	110	RS9XA	BAL
1AB2	5218	CLST 104	RS98A			RDH V U+2
1AB4	72A8	CLST 105				STH V AS,T+2
1AB6	5218	CLST 106				RDH V U+2
1AB8	72A0	CLST 107				STH V AS,T
1ABA	AD30	CLST 108		115	RS9XB	BAL
1ABC	9AB2	CLST 109		104	RS98A	BR
2D26	AB53	CLST 110	RS9XA	CICY 045	ADDERR N	BR IF H1=NZ
2D28	01C3	CLST 111				Z=U1*-KOC
2D2A	FOA4	CLST 112		063	SPEC	BR IF LZNZ
2D2C	55BB	CLST 113				T1=G1H
2D2E	128E	CLST 114				RTN
2D30	5531	CLST 115	RS9XB			V1=G1X
2D32	6351	CLST 116				V1=V1G1
2D34	C4A3	CLST 117		062	RSEND	BR IF Z=0
2D36	251D	CLST 118				G1=G1+K10

ADDR WORD SEQUENCE NO. LABEL NEXTSEQ NEXTLABEL STATEMENT

2D38 55BB CLST 119
 2D3A 128E CLST 120

T1=G1H
 RTN

 * CROSS REFERENCE FOR CSECT CLST *

CLST 005 CICY 147
 CLST 007 CLST 041
 CLST 009 CBIN 032
 CLST 011 CSFT 074
 CLST 014 CLST 012
 CLST 018 CICY 149
 CLST 019 CLST 043
 CLST 020 CLST 006 CLST 008
 CLST 024 CBIN 035 CLST 022
 CLST 025 CLST 035 CLST 037 CLST 075
 CLST 030 CICY 148
 CLST 034 CBIN 033 CLST 023
 CLST 036 CBIN 034 CLST 019 CLST 031 CLST 033
 CLST 040 CICY 150
 CLST 046 CICY 155
 CLST 055 CICY 163
 CLST 062 CLST 117
 CLST 063 CLST 057 CLST 068 CLST 112
 CLST 066 CICY 171
 CLST 074 CLST 072
 CLST 078 CICY 179
 CLST 086 CICY 187
 CLST 094 CICY 203
 CLST 095 CLST 100
 CLST 103 CICY 211
 CLST 104 CLST 109
 CLST 110 CLST 094 CLST 103
 CLST 115 CLST 099 CLST 108

CMLT -- MULTIPLY BINARY ROUTINE

DESCRIPTION

```

*-----*
*          .          .          INSTRUCTION FORMAT          *
* ENTRY  . OPERATION . BYTE 1 . BYTE 2 . BYTE 3 . BYTE 4 *
*-----*
* RROP1C . MULTIPLY   . 1C   . R1 R2 . . . . *
* RXOP4C . MULTIPLY HALFWORD. 4C . R1 X2 . B2 D2 . D2 *
* RXOP5C . MULTIPLY FULLWORD. 5C . R1 X2 . B2 D2 . D2 *
*-----*
    
```

THE ROUTINE FETCHES THE OPERANDS, THEN PERFORMS SIGN ANALYSIS. THIS RESULT IS STORED FOR USE AT THE END OF THE ROUTINE.

THE MULTIPLY TABLE IS STORED IN AUX STORAGE. TO BUILD THE TABLE, THE ROUTINE MAKES 4 PASSES THROUGH A LOOP. IN THE LOOP, THE MULTIPLICAND IS STORED, THEN DOUBLED AND REDOUBLED. THEREFORE, THE VALUES STORED IN THE TABLE ARE -- MULTIPLICAND TIMES 1, 4, 16, AND 64. SEE AUX STORAGE MAP IN SECTION AAAB FOR LOCATION.

MULTIPLICATION IS DONE ON A BYTE BASIS. FOR THE LOW DIGIT OF THE MULTIPLIER BYTE, THE TABLE ADDRESS IS INITIALIZED FOR X1. FOR THE HIGH DIGIT, THE TABLE ADDRESS IS SET FOR X16. THEN, A 16-WAY BRANCH ON THE ACTUAL MULTIPLIER DIGIT OCCURS. THE BRANCH DIRECTS THE MICROROUTINE TO THE PROPER COMPUTE ROUTINE FOR THE CURRENT MULTIPLIER DIGIT. THE COMPUTE ROUTINES CHANGE THE MULTIPLY-TABLE ADDRESS IF A X4 OR A X64 TABLE VALUE IS NEEDED. ALSO, COMPLEMENT ADD IS SET IF A SUBTRACT IS NEEDED. THEN THE MICROROUTINE IS DIRECTED TO ADD EITHER ONCE OR TWICE DEPENDING ON THE COMPUTE ROUTINE FOR THE CURRENT DIGIT. NOTE THAT FOR SOME DIGITS, ADDS FROM TWO DIFFERENT TABLE ADDRESSES ARE NECESSARY.

SHIFTING OCCURS AFTER BOTH THE HIGH AND LOW DIGITS FOR EACH MULTIPLIER BYTE ARE PROCESSED. THE CURRENTLY COMPLETED PRODUCT BYTE IS SHIFTED OUT OF THE PRODUCT-BUILDING AREA AND INTO THE REGISTERS FORMERLY OCCUPIED BY THE MULTIPLIER.

A COUNTER DETERMINES THE LAST MULTIPLIER DIGIT. AFTER THIS LAST DIGIT IS PROCESSED, A CARRY TEST IS MADE. A FINAL X1 VALUE FROM THE TABLE IS COMPLEMENT-ADDED TO THE PRODUCT IF THE NEED IS INDICATED BY THE SIGN AND MULTIPLIER-LOW DIGIT CARRY.

TO COMPLETE THE MULTIPLY OPERATION, THE MICROPROGRAM RESTORES THE SAVED REGISTERS, AND THE HALFWORD OR FULLWORD PRODUCT IS STORED AT THE 1ST OPERAND ADDRESS. THEN, THE MICROPROGRAM RETURNS TO I-CYCLES.

OBJECTIVES

1. FETCH BOTH OPERANDS. FOR FULLWORD MULTIPLY, THE MULTIPLICAND IS FOUND IN THE ODD REGISTER FOLLOWING THE EVEN REGISTER SPECIFIED BY R1. FOR MULTIPLY HALFWORD, R1 SPECIFIES THE LOCATION OF THE MULTIPLICAND. THE MULTIPLIER IS FOUND IN R2 OR AT THE EFFECTIVE MAIN STORAGE ADDRESS.
2. CREATE A TABLE OF MULTIPLICAND VALUES IN AUX STORAGE -- MULTIPLICAND TIMES 1, X4, X16, AND X64.
3. READ EACH HEXADECIMAL MULTIPLIER DIGIT, STARTING AT THE LOW-ORDER. CREATE PRODUCT BY ADDING APPROPRIATE TABLE VALUES OR COMBINATION OF VALUES AS DETERMINED BY THE MULTIPLIER DIGIT. THE PRODUCT IS SHIFTED AFTER ADDING THE TABLE VALUES FOR BOTH THE HIGH AND LOW DIGITS OF EACH MULTIPLIER BYTE.
4. STORE THE PRODUCT IN THE LOCATION THAT ORIGINALLY CONTAINED THE MULTIPLICAND.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CMLT 001	T			MULTIPLY BINARY OP CODES	D. L. SMITH
0784	F503	CMLT 002	RXOP5C	006	ERRORA	BR IF G13=1	
0786	3515	CMLT 003				G1=G1\$K10	
0788	A770	CMLT 004		CCOM 045	CXGET	BAL	
078A	82FA	CMLT 005		015	START	BR	
0782	AE98	CMLT 006	ERRORA	CICY 059	SPECHK	BR	
078C	E191	CMLT 007	RROP1C	009	NECLAB	BR IF S6=1	
078E	3485	CMLT 008				GO=GO\$K80	
0790	F503	CMLT 009	NEQLAB	006	ERRORA	BR IF G13=1	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT
0792	3515	CMLT 010				G1=G1\$K10
0794	8A70	CMLT 011		CCOM 036	CRGET	BAL
0796	82FA	CMLT 012		015	START	BR
		CMLT 013	ASEQ	AL07=78		
02F8	ADA8	CMLT 014	RXOP4C	CCOM 012	CXGETH	BAL
02FA	8DC2	CMLT 015	START	CCOM 057	LSAVEA	BAL
02FC	29C7	CMLT 016				I1=0
02FE	C604	CMLT 017		020	CANDPL	BR IF D00=0
0300	29F7	CMLT 018				I1=0\$KFF
0302	2C13	CMLT 019				PO=0\$K01
0304	C275	CMLT 020	CANDPL			Z=V0*-K70
0306	EC8B	CMLT 021		023	PLIERP	BR IF HZ=0
0308	3C23	CMLT 022				PO=PO\$K02
030A	5989	CMLT 023	PLIERP			I0=I1
030C	4A86	CMLT 024				T=I
030E	00C0	CMLT 025				RST S K=0C
0310	7CF2	CMLT 026	LOOP1			STH P DA, BE
0312	4C52	CMLT 027	FLTMU1			RDH P DC, 9A
0314	2D1D	CMLT 028	LUOP			PI=P1+K10
0316	2D8B	CMLT 029				PI=PI+K08
0318	CD29	CMLT 030		038	EXIT	BR IF P10=1
031A	78C8	CMLT 031				STH I AS, P+2
031C	7AC8	CMLT 032				STH T AS, P+2
031E	76C8	CMLT 033				STH D AS, P+2
0320	7EC8	CMLT 034				STH H AS, P+2
0322	856E	CMLT 035		100	DOUBLE	BAL
0324	856E	CMLT 036		100	DOUBLE	BAL
0326	8314	CMLT 037		028	LOOP	BR
0328	2DC7	CMLT 038	EXIT			PI=0
032A	48C6	CMLT 039				I=P
032C	4AC6	CMLT 040				T=P
032E	44C6	CMLT 041				G=P
0330	46C6	CMLT 042				D=P
0332	4EC6	CMLT 043				H=U
0334	40C6	CMLT 044				U=P
0336	2C13	CMLT 045				PO=0\$K01
0338	C5BE	CMLT 046	CHKLO4	049	LOCHK	BR IF S0=0
033A	2F1B	CMLT 047	NYX			H1=H1+K01
033C	F4D3	CMLT 048		059	ADJUST	BR IF AC=1
033E	2D15	CMLT 049	LOCHK			PI=0\$K10
0340	1002	CMLT 050				RST S K=90
0342	BB55	CMLT 051		130	DIG N	N=H1L
0344	D053	CMLT 052	CYCLE	059	ADJUST	BR IF P11=1
0346	C5CC	CMLT 053		056	HICLK	BR IF S0=0
0348	2F1D	CMLT 054				H1=H1+K10
034A	F4D3	CMLT 055		059	ADJUST	BR IF AC=1
034C	2D55	CMLT 056	HICLK			PI=0\$K50
034E	1002	CMLT 057				RST S K=90
0350	BF55	CMLT 058		130	DIG N	N=H1H
0352	2C2B	CMLT 059	ADJUST			PO=PO+K02
0354	9468	CMLT 060		081	ROTA	BAL
0356	C838	CMLT 061	FLTMU2	046	CHKLO4	BR IF P04=0
0358	C1DC	CMLT 062	ALMOSA	064	ALMOST	BR IF S4=0

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
035A	95F6	CMLT 063		CFMU 009	MULT	BR	EXIT TO FLOAT PT MULT
035C	54F2	CMLT 064	ALMOST			RDH G DA, BE	
035E	C5E9	CMLT 065		070	ANSWER	BR IF S0=1	
0360	E068	CMLT 066		070	ANSWER	BR IF G06=0	
0362	30C2	CMLT 067				SET S K=90	
0364	2D15	CMLT 068				P1=0\$K10	
0366	9CD2	CMLT 069		109	ADD1	BAL	
0368	9256	CMLT 070	ANSWER	CCOM 065	LRSTRA	BAL	
036A	2AC7	CMLT 071				T0=0	
036C	55BB	CMLT 072				T1=G1H	
036E	72A8	CMLT 073				STH V AS, T+2	
0370	7EAA	CMLT 074				STH H AS, T-2	
0372	F47A	CMLT 075		079	DONE	BR IF G03=0	
0374	1B15	CMLT 076				T1=T1*-K10	
0376	76A8	CMLT 077				STH D AS, T+2	
0378	70AA	CMLT 078				STH U AS, T-2	
037A	A8DE	CMLT 079	DONE	CICY 004	CHECK	BR	
		CMLT 080	AEND				
1468	5EF9	CMLT 081	ROTA			H1=H0	
146A	53E9	CMLT 082				H0=V1	
146C	5239	CMLT 083				V1=V0	
146E	5129	CMLT 084				V0=U1	
1470	5019	CMLT 085				U1=U0	
1472	5709	CMLT 086				U0=D1	
1474	5679	CMLT 087				D1=D0	
1476	5B69	CMLT 088				D0=T1	
1478	5AB9	CMLT 089				T1=T0	
147A	55A9	CMLT 090				T0=G1	
147C	5459	CMLT 091				G1=GO	
147E	24C7	CMLT 092				GO=0	
1480	C50C	CMLT 093		099	NOFLOW	BR IF G10=0	
1482	058F	CMLT 094				Z=G1\$K88	
1484	E08A	CMLT 095		098	FLCFIX	BR IF HZNZ	
1486	C10C	CMLT 096		099	NOFLOW	BR IF G14=0	
1488	C58C	CMLT 097		099	NOFLOW	BR IF S0=0	
148A	24F7	CMLT 098	FLOFIX			GO=0\$KFF	
148C	128E	CMLT 099	NOFLOW			RTN	
056E	6FFB	CMLT 100	DOUBLE			H1C=H1+H1	
0570	6EED	CMLT 101				H0C=H0+H0+C	
0572	677D	CMLT 102				D1C=D1+D1+C	
0574	666D	CMLT 103				D0C=D0+D0+C	
0576	6BBD	CMLT 104				T1C=T1+T1+C	
0578	6AAD	CMLT 105				T0C=T0+T0+C	
057A	699D	CMLT 106				I1C=I1+I1+C	
057C	688D	CMLT 107				I0C=I0+I0+C	
057E	128E	CMLT 108				RTN	
1CD2	1CFB	CMLT 109	ADD1			P0=P0\$K0F	
1CD4	3DE3	CMLT 110	ADDR0U			P1=P1\$K0E	
1CD6	58CA	CMLT 111				RDH I AS, P-2	
1CD8	719D	CMLT 112				U1C=U1\$11+C	
1CDA	708D	CMLT 113				U0C=U0\$10+C	
1CDC	58CA	CMLT 114				RDH I AS, P-2	
1CDE	779D	CMLT 115				D1C=D1\$11+C	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1CE0	768D	CMLT 116				DOC=D0%I0+C	
1CE2	58CA	CMLT 117				RDH I AS,P-2	
1CE4	7B9D	CMLT 118				T1C=T1%I1+C	
1CE6	7A8D	CMLT 119				T0C=T0%I0+C	
1CE8	58CA	CMLT 120				RDH I AS,P-2	
1CEA	759D	CMLT 121				G1C=G1%I1+C	
1CEC	748D	CMLT 122				G0C=G0%I0+C	
1CEE	1CFB	CMLT 123				P0=P0%K0F	
1CF0	0002	CMLT 124				RST S3	
1CF2	F87B	CMLT 125		129	OUTOUT	BR IF P07=1	
1CF4	C5D4	CMLT 126		110	ADDR0U	BR IF S0=0	
1CF6	2002	CMLT 127				SET S3	
1CF8	C5D5	CMLT 128		110	ADDR0U	BR IF S0=1	
1CFA	128E	CMLT 129	OUTOUT			RTN	
02C0	8344	CMLT 130	DIG 0	052	CYCLE	BR	0
02C2	82E0	CMLT 131	DIG 1	146	PUTE1	BR	+16 OR 1
02C4	BC78	CMLT 132	DIG 2	163	PUTE2	BR	+32 OR 2
02C6	B0EE	CMLT 133	DIG 3	148	PUTE3	BR	+48 OR 3
02C8	B0F4	CMLT 134	DIG 4	151	PUTE4	BR	+64 OR 4
02CA	B0F0	CMLT 135	DIG 5	149	PUTE5	BR	+80 OR 5
02CC	88E2	CMLT 136	DIG 6	154	PUTE6	BR	+96 OR 6
02CE	B070	CMLT 137	DIG 7	159	PUTE7	BR	+112 OR 7
02D0	BC76	CMLT 138	DIG 8	162	PUTE8	BR	+128 OR 8
02D2	B34E	CMLT 139	DIG 9	165	PUTE9	BR	-112 OR 7
02D4	A36E	CMLT 140	DIG A	168	PUTEA	BR	-96 OR 6
02D6	8390	CMLT 141	DIG B	177	PUTEB	BR	-80 OR 5
02D8	87FC	CMLT 142	DIG C	175	PUTEC	BR	-64 OR 4
02DA	8392	CMLT 143	DIG D	178	PUTED	BR	-48 OR 3
02DC	A376	CMLT 144	DIG E	172	PUTEE	BR	-32 OR 2
02DE	3002	CMLT 145	DIG F			SET S K=90	-16 OR 1
02E0	9CD2	CMLT 146	PUTE1	109	ADD1	BAL	
02E2	8344	CMLT 147		052	CYCLE	BR	
30EE	3002	CMLT 148	PUTE3			SET S K=90	
30F0	9CD2	CMLT 149	PUTE5	109	ADD1	BAL	
30F2	1002	CMLT 150				RST S K=90	
30F4	3025	CMLT 151	PUTE4			P1=P1%K20	
30F6	9CD2	CMLT 152		109	ADD1	BAL	
30F8	8344	CMLT 153		052	CYCLE	BR	
08E2	3D25	CMLT 154	PUTE6			P1=P1%K20	
08E4	9CD2	CMLT 155		109	ADD1	BAL	
08E6	1D25	CMLT 156				P1=P1*-K20	
08E8	9CD4	CMLT 157		110	ADDR0U	BAL	
08EA	8344	CMLT 158		052	CYCLE	BR	
3070	3002	CMLT 159	PUTE7			SET S K=90	
3072	9CD2	CMLT 160		109	ADD1	BAL	
3074	1002	CMLT 161				RST S K=90	
3076	3D25	CMLT 162	PUTE8			P1=P1%K20	
3078	9CD4	CMLT 163	PUTE2	110	ADDR0U	BAL	
307A	8344	CMLT 164		052	CYCLE	BR	
334E	9CD2	CMLT 165	PUTE9	109	ADD1	BAL	
3350	3002	CMLT 166				SET S K=90	
3352	B076	CMLT 167		162	PUTE8	BR	
236E	3002	CMLT 168	PUTEA			SET S K=90	

* CROSS REFERENCE FOR CSECT CMLT *

CMLT 172	CMLT 144
CMLT 175	CMLT 142
CMLT 177	CMLT 141
CMLT 178	CMLT 143

CMPU DESCRIPTIVE TEXT

LABEL..SSOPF1 MOVE WITH OFFSET

MOVE WITH OFFSET EXAMPLE

STEP

- 1..READ LOW SOURCE BYTE
- 2..MOVE LOW 4 BITS OF SOURCE BYTE TO HIGH 4 BITS OF THE LOW DESTINATION BYTE. THE LOW 4 BITS OF THE DESTINATION BYTE REMAIN UNCHANGED.
- 3..CHECK TO SEE WHAT FIELD HAS ENDED,
 - A. IF DESTINATION OR BOTH ENDED, RETURN TO I-CYCLES.
 - B. IF SOURCE HAS ENDED, STORE THE HIGH 4 BITS OF THE SOURCE BYTE IN THE LOW 4 BITS OF THE NEXT DESTINATION BYTE. FILL DESTINATION REMAINDER WITH ZEROES, RETURN TO I-CYCLES.
 - C. IF NEITHER HAS ENDED, READOUT NEXT SOURCE BYTE, STORE THE HIGH 4 BITS OF PREVIOUS SOURCE BYTE AS LOW 4 BITS IN DESTINATION BYTE. STORE LOW 4 BITS OF NEW SOURCE BYTE AS HIGH 4 BITS IN DESTINATION BYTE. RETURN TO STEP 3.

```

DESTINATION BEFORE 79 56 87 84 3C
SOURCE              64 72 63 75 91 85 29 76
                   .
                   .....
                   .
                   .
                   .
                   .
                   .....
DESTINATION AFTER 59 18 52 97 6C
    
```

LABEL..SSOPF2 PACK

PACK EXAMPLE

STEP

- 1..READOUT LOW SOURCE BYTE. CROSS DIGITS AND STORE AT LOW DESTINATION ADDRESS.
- 2..CHECK TO SEE WHAT FIELD HAS ENDED.
 - A. IF DESTINATION OR BOTH ENDED, RETURN TO I-CYCLES.
 - B. IF SOURCE HAS ENDED, FILL REMAINING DESTINATION WITH ZEROES.
 - C. IF NEITHER HAS ENDED, READOUT NEXT SOURCE BYTE. IF THIS BYTE ENDS SOURCE FIELD, FILL REMAINING DESTINATION WITH ZEROES, AFTER STORING LOW 4 SOURCE BITS. IF SOURCE DID NOT END, READOUT NEXT SOURCE BYTE, COMBINE LOW ORDER SOURCE DIGITS AND STORE AT NEXT HIGHER DESTINATION ADDRESS. RETURN TO STEP 2.

```

SOURCE              F6 F7 F2 F4 F9 C3
DESTINATION BEFORE 00 C0 00 00 00
DESTINATION AFTER  00 06 72 49 3C
    
```

LABEL..SSOPF3 UNPACK

UNPACK EXAMPLE

THE UNPACK INSTRUCTION IS ALMOST THE REVERSE OF PACK. PROPER ZONES ARE INSERTED IN DESTINATION BYTES FOR ALL SOURCE DIGITS. ZONES ARE INSERTED WITH ZEROES WHEN FILLING IN DESTINATION FIELD WHEN SOURCE HAS ENDED.

```

SOURCE              64 92 5C
DESTINATION BEFORE 00 00 00 00 00 00
DESTINATION AFTER  F0 F6 F4 F9 F2 C5
    
```


ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CMPU 001	T			PACK,UNPACK, AND MOVE WITH OFFSET	12/15/66 R. C. HUANG
		CMPU 002	*				
0186	5B3A	CMPU 003	SSOPF1			RDB T1 V-1	MOVE OFFSET, READ SOURCE SIGN
0188	5F10	CMPU 004				RDB H1 U+0	READ DESTINATION SIGN BYTE
018A	4BF3	CMPU 005				H1=T1XH+H1L	SET UP THE SIGN BYTE
018C	5B75	CMPU 006				D1=T1XL	SAVE HI HALF BYTE OF SOURCE
018E	AE88	CMPU 007		013	STBYTE	BR	
2E7E	27F5	CMPU 008	SSOPF3			D1=0\$KFO	UNPACK, SET UP FOR EBCDIC MODE
2E80	C904	CMPU 009		011	SSOPF2	BR IF P1 BIT4=0	BR ON NOT ASCII MODE
2E82	2755	CMPU 010				D1=0\$K50	SET UP FOR ASCII MODE
2E84	5B3A	CMPU 011	SSOPF2			RDB T1 V-1	PACK, READ SIGN BYTE
2E86	5BF1	CMPU 012				H1=T1X	CROSS SIGN BYTE
2E88	7F1A	CMPU 013	STBYTE			STB H1 U-1	STORE SIGN BYTE
2E8A	5559	CMPU 014				G1=G1	PUT LENGTH BYTE ON Z-BUSS
2E8C	8083	CMPU 015		016	HZLZBR N	N=DYN BITS67	BR ON HZ,LZ - ZERO LENGTH
2900	8023	CMPU 016	HZLZBR 0	020	OP N	N=GO BITS67	BR ON OP'S
2902	A932	CMPU 017	HZLZBR 1	028	STDEST	BR	SOURCE ENDED
2904	81B3	CMPU 018	HZLZBR 2	CICY 005	ISTART N	N=S BITS67	DESTINATION ENDED
2906	81B3	CMPU 019	HZLZBR 3	CICY 005	ISTART N	N=S BITS67	BOTH ENDED
2922	81A6	CMPU 020	OP 1	032	MWOS	BR	MOVE WITH OFFSET
2924	AE12	CMPU 021	OP 2	037	PACK	BR	PACK
2926	5B3A	CMPU 022	OP 3			RDB T1 V-1	UNPACK, READ SOURCE
2928	4B7D	CMPU 023				D1=T1L+D1H	UNPACK LO HALF BYTE
292A	771A	CMPU 024				STB D1 U-1	STORE FIRST BYTE
292C	351D	CMPU 025				G1=G1-K10	SUBTRACT HI, LO COUNT BY 1
292E	E087	CMPU 026		019	HZLZBR 3	BR IF HZ=0	BR IF DESTINATION END
2930	4B75	CMPU 027				D1=T1XL\$D1H	UNPACK HI HALF BYTE
2932	771A	CMPU 028	STDEST			STB D1 U-1	STORE DESTINATION BYTE
2934	577B	CMPU 029				D1=D1H	ZERO OUT NUMERIC
2936	25FD	CMPU 030				G1=G1+KFO	SUBTRACT DEST COUNT
2938	8083	CMPU 031		016	HZLZBR N	N=DYN BITS67	BR ON HZ,LZ COUNT
01A6	5B3A	CMPU 032	MWOS			RDB T1 V-1	READ SOURCE
01A8	4B73	CMPU 033				D1=T1XH+D1L	SET UP THE BYTE
01AA	771A	CMPU 034				STB D1 U-1	STORE DESTINATION
01AC	5B75	CMPU 035				D1=T1XL	SAVE HI HALF BYTE FOR NEXT CYCLE
01AE	AE1E	CMPU 036		043	HILOCT	BR	GO TO SUBT HI, LO COUNT
2E12	5F3A	CMPU 037	PACK			RDB H1 V-1	READ FIRST SOURCE BYTE
2E14	25FF	CMPU 038				G1=G1+KFF	SUBTRACT LOW COUNT BY 1
2E16	FC8F	CMPU 039		045	SOUEND	BR IF LZ=0	BR IF SOURCE ENDED
2E18	5B3A	CMPU 040				RDB T1 V-1	READ SECOND SOURCE BYTE
2E1A	4BF3	CMPU 041				H1=T1XH+H1L	PACK THE TWO BYTES
2E1C	7F1A	CMPU 042				STB H1 U-1	STORE IT
2E1E	351D	CMPU 043	HILOCT			G1=G1-K10	SUBT HI, LO COUNT BY 1
2E20	8083	CMPU 044		016	HZLZBR N	N=DYN BITS67	BR ON HZ, LZ COUNT
2E0E	5F7D	CMPU 045	SOUEND			D1=H1L	SOURCE ENDED, ZERO OUT HI HALF
2E10	A932	CMPU 046		028	STDEST	BR	GO TO STORE & BR HZ

 * CROSS REFERENCE FOR CSECT CMPU *

CMPU 003 CICY 242
 CMPU 008 CICY 244
 CMPU 011 CICY 243 CMPU 009
 CMPU 013 CMPU 007

* CROSS REFERENCE FOR CSECT CMPU *

CMPU 016	CMPU 015	CMPU 031	CMPU 044
CMPU 019	CMPU 026		
CMPU 020	CMPU 016		
CMPU 028	CMPU 017	CMPU 046	
CMPU 032	CMPU 020		
CMPU 037	CMPU 021		
CMPU 043	CMPU 036		
CMPU 045	CMPU 039		

CNVR DESCRIPTIVE TEXT

-----						OBJECTIVES
* ENTRY *	* OPERATION *	* BYTE 1 *	* BYTE 2 *	* BYTE 3 *	* BYTE 4 *	

* RXOP4E *	* CONVERT TO DEC. *	* 4E *	* R1 X2 *	* B2 D2 *	* D2 *	CONVERT TO BINARY - THE SECOND OPERAND IS CHANGED FROM A PACKED DECIMAL NUMBER TO A BINARY NUMBER AND PLACED IN THE GENERAL PURPOSE REGISTER SPECIFIED BY R1. THE SECOND OPERAND IS CHECKED FOR VALID SIGN AND DIGIT CODES.
* RXOP4E *	* CONVERT TO BIN. *	* 4F *	* R1 X2 *	* B2 D2 *	* D2 *	

PRIOR TO ENTRY TO THIS ROUTINE, THE EFFECTIVE ADDRESS FOR THE SECOND OPERAND HAS BEEN CALCULATED IN THE CICY ROUTINE.

CONVERT TO DECIMAL - THE FIRST OPERAND IS CHANGED FROM A SIGNED BINARY NUMBER TO A PACKED DECIMAL NUMBER AND PLACED IN THE SECOND OPERAND LOCATION.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CNVR 001	T			CONVERT TO DECIMAL AND BINARY ROUTINES	
		CNVR 002	*				
		CNVR 003	*				
		CNVR 004	*			CONVERT TO DECIMAL	
29C4	AB53	CNVR 005	RXOP4E	CICY 045	ADDERR N	BR IF H1=NZ	
29C6	0183	CNVR 006				Z=U1*-K08	
29C8	FOCD	CNVR 007		009	OKAY	BR IF LZ=0	
29CA	AE98	CNVR 008		CICY 059	SPECHK	BR	
29CC	8DC2	CNVR 009	OKAY	CCOM 057	LSAVEA	BAL	
29CE	10CE	CNVR 010				RST S K=FC	
29D0	FC54	CNVR 011		013	RX4EA	BR IF G07=0	
29D2	AAFA	CNVR 012		058	RXOP4F	BR	
29D4	ADB8	CNVR 013	RX4EA	CCOM 020	GETR1	BAL	
29D6	2B07	CNVR 014				T1=0	STORE FIRST
29D8	7A18	CNVR 015				STH T U+2	TWO ANS. BYTES
29DA	C660	CNVR 016		019	RX4EB	BR IF D00=0	
29DC	8AE0	CNVR 017		CCOM 076	COMPL	BAL	
29DE	2040	CNVR 018				SET S5	SET MINUS
29E0	42A6	CNVR 019	RX4EB			V=T	
29E2	44A6	CNVR 020				G=T	
29E4	48A6	CNVR 021				I=T	
29E6	2D13	CNVR 022				P1=0\$K01	
29E8	C66F	CNVR 023		026	L4EB	BR IF D00=1	
29EA	AD4E	CNVR 024	L4EA	039	DOBBIN	BAL	
29EC	C66A	CNVR 025		024	L4EA	BR IF D00=0	
29EE	AD3C	CNVR 026	L4EB	030	RX4EAA	BAL	
29F0	C66F	CNVR 027	L4EC	026	L4EB	BR IF D00=1	
29F2	AD3E	CNVR 028		031	RX4EAB	BAL	
29F4	A9F0	CNVR 029		027	L4EC	BR	
203C	3915	CNVR 030	RX4EAA			I1=I1\$K10	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2D3E	ED5D	CNVR 031	RX4EAB	046	END	BR IF P12=1	
2D40	C002	CNVR 032				RST S3	
2D42	799F	CNVR 033				I1C=I1@I1+C	
2D44	788F	CNVR 034				I0C=I0@I0+C	
2D46	755F	CNVR 035				G1C=G1@G1+C	
2D48	744F	CNVR 036				G0C=G0@G0+C	
2D4A	733F	CNVR 037				V1C=V1@V1+C	
2D4C	722F	CNVR 038				V0C=V0@V0+C	
2D4E	6FFB	CNVR 039	DOBBIN			H1C=H1+H1	
2D50	6EED	CNVR 040				H0C=H0+H0+C	
2D52	677D	CNVR 041				D1C=D1+D1+C	
2D54	666D	CNVR 042				D0C=D0+D0+C	
2D56	ED5D	CNVR 043		046	END	BR IF P12=1	
2D58	2D1B	CNVR 044				P1=P1+K01	
2D5A	128E	CNVR 045				RTN	
2D5C	4686	CNVR 046	END			D=I	
2D5E	9258	CNVR 047		CCOM 066	LRSTRB	BAL	
2D60	27CB	CNVR 048				D1=D1+K0C	SET NOT ASCII PLUS
2D62	C966	CNVR 049		051	NEXTCK	BR IF P14=0	
2D64	176B	CNVR 050				D1=D1@K06	SET ASCII PLUS
2D66	D1EA	CNVR 051	NEXTCK	053	FINAL	BR IF S5=0	
2D68	3713	CNVR 052				D1=D1\$K01	MAKE SIGN MINUS
2D6A	7218	CNVR 053	FINAL			STH V U+2	STORE
2D6C	7418	CNVR 054				STH G U+2	RESULT
2D6E	7618	CNVR 055				STH D U+2	
2D70	81B3	CNVR 056		CICY 005	ISTART N	N=S BITS67	
		CNVR 057	*			CONVERT TO BINARY	
2AFA	2B83	CNVR 058	RXOP4F			T1=0\$K08	STORE
2AFC	3B15	CNVR 059				T1=T1\$K10	THE
2AFE	4EA6	CNVR 060				H=T	DECIMAL
2B00	5718	CNVR 061	LOOPAX			RDB D1 U+1	FIELD
2B02	77EF	CNVR 062				D1C=D1@H0+C	TN
2B04	E589	CNVR 063		065	NOTSAV	BR IF S2=1	AUX
2B06	5AE4	CNVR 064				T=H+1	STORAGE
2B08	77E8	CNVR 065	NOTSAV			STB D1 AS,H+1	AND
2B0A	0FFB	CNVR 066				Z=H1@K0F	FIND
2B0C	FC80	CNVR 067		061	LOGPAX	BR IF LZNZ	1ST
2B0E	5710	CNVR 068				RDB D1 U	SIG
2B10	573B	CNVR 069				V1=D1H	DIGET
2B12	A4EC	CNVR 070		CDMD 203	SINSET	BAL	GO CHECK SIGN
2B14	73AF	CNVR 071				V1C=V1@T0+C	
2B16	C19A	CNVR 072		074	OKFLD	BR IF S4=0	
2B18	B07C	CNVR 073		CCOM 071	FLGDEX	BR	GO FLAG DATA EXCEP
2B1A	73E0	CNVR 074	OKFLD			STB V1 AS,H	
2B1C	555B	CNVR 075				G1=G1H	BUILD
2B1E	3533	CNVR 076				G1=G1\$K03	ANS
2B20	10CE	CNVR 077				RST S K=FC	REG ADDRESS
2B22	E426	CNVR 078		080	PLSNUM	BR IF G02=0	BR IF PLUS
2B24	3002	CNVR 079				SET S K=90	
2B26	2407	CNVR 080	PLSNUM			G0=0	
2B28	29C7	CNVR 081	CONTUE			I1=0	
2B2A	0040	CNVR 082				RST S5	INITIALIZE ZERO FLAG
2B2C	4EA6	CNVR 083				H=T	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2B2E	2207	CNVR 084				VO=0	
2B30	5DE0	CNVR 085	RDBP1H			RDB P1 AS,H	
2B32	5DC5	CNVR 086				PO=PLXL	
2B34	8270	CNVR 087		123	X1OPLS	BAL	
2B36	527B	CNVR 088				D1=VOH	
2B38	0FFB	CNVR 089				Z=H10KOF	
2B3A	F0E8	CNVR 090		100	LAB6	BR IF LZNZ	
2B3C	77E0	CNVR 091				STB D1 AS,H	
2B3E	14FF	CNVR 092				GO=GO0KFF	
2B40	C4CA	CNVR 093		098	LAB9	BR IF ZNZ	
2B42	4203	CNVR 094				U0=VOXH+UOL	
2B44	79CD	CNVR 095				I1C=I1%U0+C	
2B46	794A	CNVR 096				STB I1 AS,G-1	
2B48	C14F	CNVR 097		110	DONENO	BR IF G14=1	
2B4A	520D	CNVR 098	LAB9			U0=VOL	
2B4C	AB28	CNVR 099		081	CONTUE	BR	
2B68	5DCD	CNVR 100	LAB6			PO=P1L	
2B6A	8270	CNVR 101		123	X1OPLS	BAL	
2B6C	4275	CNVR 102				D1=VOXL\$D1H	
2B6E	D1F5	CNVR 103		106	LAB7	BR IF S5=1	BR IF NZ RESULT SO FAR
2B70	C4F8	CNVR 104		108	LAWOPS	BR IF ZNZ	SET NOT ZERO FLAG
2B72	5AA4	CNVR 105				T=T+1	
2B74	77E8	CNVR 106	LAB7			STB D1 AS,H+1	
2B76	AB30	CNVR 107		085	RDBP1H	BR	
2B78	2040	CNVR 108	LAWOPS			SET S5	
2B7A	AB74	CNVR 109		106	LAB7	BR	
2B4E	59F9	CNVR 110	DONENO			H1=I1	
2B50	9256	CNVR 111		CCOM 065	LRSTRA	BAL	
2B52	C5DA	CNVR 112		116	LAB12	BR IF S0=0	
2B54	E5DC	CNVR 113		117	LAB13	BR IF S2=0	
2B56	CF62	CNVR 114		120	LAB14	BR IF H10=0	
2B58	AB5C	CNVR 115		117	LAB13	BR	
2B5A	CF63	CNVR 116	LAB12	120	LAB14	BR IF H10=1	
2B5C	47BB	CNVR 117	LAB13			T1=D1H+T1L	
2B5E	0BFB	CNVR 118				Z=T10KOF	
2B60	C4E7	CNVR 119		122	LAB11	BR IF Z=0	
2B62	2B93	CNVR 120	LAB14			T1=0\$K09	
2B64	9C70	CNVR 121		BPSW 003	PROG	BR	
2B66	81B3	CNVR 122	LAB11	CICY 005	ISTART N	N=S BITS67	
0270	522D	CNVR 123	X1OPLS			VO=VOL	
0272	6223	CNVR 124				VO=VO+VO	X2
0274	5239	CNVR 125				V1=VO	
0276	6223	CNVR 126				VO=VO+VO	X4
0278	6223	CNVR 127				VO=VO+VO	X8
027A	6233	CNVR 128				VO=VO+V1	X10
027C	62C3	CNVR 129				VO=VO+PO	PLUS ADDATIVE
027E	128E	CNVR 130				RTN	

* CROSS REFERENCE FOR CSECT CNVR *

CNVR 005	CICY 177	CICY 178
CNVR 009	CNVR 007	
CNVR 013	CNVR 011	
CNVR 019	CNVR 016	
CNVR 024	CNVR 025	
CNVR 026	CNVR 023	CNVR 027
CNVR 027	CNVR 029	
CNVR 030	CNVR 026	
CNVR 031	CNVR 028	
CNVR 039	CNVR 024	
CNVR 046	CNVR 031	CNVR 043
CNVR 051	CNVR 049	
CNVR 053	CNVR 051	
CNVR 058	CNVR 012	
CNVR 061	CNVR 067	
CNVR 065	CNVR 063	
CNVR 074	CNVR 072	
CNVR 080	CNVR 078	
CNVR 081	CNVR 059	
CNVR 085	CNVR 107	
CNVR 098	CNVR 093	
CNVR 100	CNVR 090	
CNVR 106	CNVR 103	CNVR 109
CNVR 108	CNVR 104	
CNVR 110	CNVR 097	
CNVR 116	CNVR 112	
CNVR 117	CNVR 113	CNVR 115
CNVR 120	CDVD 099	CNVR 114 CNVR 116
CNVR 122	CNVR 119	
CNVR 123	CNVR 087	CNVR 101

CSAS -- DECIMAL ROUTINES

ENTRIES

1. SSOPF8 -- ZERO AND ADD
2. SSOPF9 -- COMPARE DECIMAL
3. SSOPFA -- ADD DECIMAL
4. SSOPFB -- SUBTRACT DECIMAL

OBJECTIVES

DECIMAL ADD OR SUBTRACT

THE 2ND OPERAND, CALLED THE SOURCE FIELD, IS EITHER TRUE ADDED OR COMPLEMENT ADDED TO THE 1ST OPERAND, WHICH IS CALLED THE DESTINATION FIELD. THE OPERAND SIGNS AND OP CODE DETERMINE WHETHER TRUE OR COMPLEMENT ADD IS PERFORMED AS FOLLOWS.

```

*****
*          SIGN          .      TRUE OR COMPLEMENT      *
*-----*-----*-----*-----*
* 1ST OPERAND . 2ND OPERAND .   ADD OP   . SUBTRACT OP *
*-----*-----*-----*-----*
*   PLUS      .   PLUS      .  TRUE      .  COMPLEMENT *
*  MINUS      .   PLUS      .  COMPLEMENT .  TRUE      *
*  MINUS      .  MINUS      .  TRUE      .  COMPLEMENT *
*   PLUS      .  MINUS      .  COMPLEMENT .  TRUE      *
*****
    
```

ZERO AND ADD

THE 2ND OPERAND IS ADDED TO A VALUE OF ZERO AND PLACED IN THE 1ST OPERAND LOCATION.

COMPARE DECIMAL

THE 2ND OPERAND IS COMPARED WITH THE 1ST OPERAND, AND THE RESULT OF THE COMPARISON IS INDICATED BY THE CONDITION CODE. COMPARISON IS RIGHT TO LEFT.

DESCRIPTION

- THE ROUTINE PERFORMS THE DECIMAL OPERATION AS FOLLOWS
1. FOR ADD OR ZERO AND ADD, SET TRUE ADD.
FOR SUBTRACT OR COMPARE, SET COMPLEMENT ADD.
 2. READ LOW-ORDER BYTE OF 2ND OPERAND. TEST SIGN. IF SIGN IS INVALID, INITIATE PROGRAM INTERRUPT.
 3. LOOP BACK AND TEST SIGN OF 1ST OPERAND. THIS STEP IS NOT PERFORMED FOR ZERO AND ADD.
 4. ADD OR COMPLEMENT ADD 1ST DIGIT.
 5. SET PROPER SIGN. STORE 1ST BYTE AT 1ST OPERAND, DESTINATION FIELD, THEN STEP TO NEXT DESTINATION BYTE. RESULT IS NOT STORED IF COMPARE OP.

DESCRIPTION CONTINUED

6. READ NEXT SOURCE BYTE FROM 2ND OPERAND.
READ NEXT DESTINATION BYTE FROM 1ST OPERAND.
IF ZERO AND ADD, DESTINATION FIELD IS SET TO ZERO.
7. ADD OR COMPLEMENT ADD THE TWO BYTES.
8. STORE RESULT AT 1ST OPERAND DESTINATION FIELD, AND STEP TO NEXT DESTINATION BYTE. RESULT IS NOT STORED IF COMPARE OP.
9. DECREMENT LENGTH.
10. REPEAT STEPS 6 THROUGH 9 UNTIL EITHER DESTINATION OR SOURCE FIELD ENDS.

SOURCE FIELD ENDS BEFORE DESTINATION FIELD

SUPPLY ZEROS AS SOURCE DIGITS AND CONTINUE UNTIL DESTINATION FIELD ENDS.

DESTINATION FIELD ENDS BEFORE SOURCE FIELD

READ OUT REMAINING SOURCE DIGITS. ADD THEM TO ZERO AFTER CHECKING TRUE/COMPLEMENT STATUS AND THE CARRY FROM LAST ADD AS FOLLOWS.

1. COMPLEMENT ADD AND CARRY. IF LAST ADD BEFORE DESTINATION FIELD ENDED WAS ZERO, FURTHER RESULTS MUST BE ZERO, OR AN OVERFLOW IS INDICATED.
IF PREVIOUS RESULT WAS NOT ZERO, FURTHER RESULTS MUST BE NINES.
2. COMPLEMENT ADD AND NO CARRY. RESULTS MUST BE NINES, OR AN OVERFLOW IS INDICATED.
3. TRUE ADD AND CARRY. INDICATE OVERFLOW.
4. TRUE ADD AND NO CARRY. INDICATE OVERFLOW IF FURTHER RESULTS ARE NOT ZERO.

BOTH SOURCE AND DESTINATION FIELD ENDED

1. IF INVALID DATA, INITIATE PROGRAM INTERRUPT.
2. RECOMPLEMENT RESULT IF THE FOLLOWING CONDITIONS ARE PRESENT WHEN FIELDS END.
 - A. COMPLEMENT ADD, NO CARRY, AND NOT A COMPARE OP.
 - B. TRUE ADD, NO CARRY, ALU ZERO, MINUS SIGN STATUS, AND NOT A COMPARE OP.
 RESULT IS RECOMPLEMENTED, STARTING AT LOW-ORDER BYTE. SIGN OF RESULT IS INVERTED.
3. SET CONDITION CODES AND RETURN TO I-CYCLES.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CSAS 001	T			DECIMAL ADD, SUBTRACT, COMPARE, AND ZERO AND ADD	
		CSAS 002	*				
		CSAS 003	*			OPERAND 1 ADDRESS IS IN U H1 IS ZERO TO IS ZERO	
		CSAS 004	*			OPERAND 2 ADDRESS IS IN V D1 IS ZERO	
		CSAS 005	ASEQ			AL07=4C	
114C	3443	CSAS 006	SSOPF8			GO=GO\$K04	ZERO AND ADD
114E	1082	CSAS 007	SSOPFA			RST S K=98	RESET S0,S3,S4 ADD
1150	2E15	CSAS 008	STUPDC			HO=0\$K10	SET UP DECREMENT
1152	573A	CSAS 009				RDB D1 V-1	READ 1ST 2ND OP BYTE
1154	576D	CSAS 010	TSTSGN			DO=D1L	
1156	0661	CSAS 011				Z=DO+K06	
1158	E0F3	CSAS 012		025	NOSIGN	BR IF HZ=0	
115A	00C4	CSAS 013				RST S K=2C	RESET S2,S4,S5
115C	F366	CSAS 014		019	ISPLUS	BR IF D1 BIT7=0	TEST FOR PLUS A,C,E
115E	07FB	CSAS 015				Z=D1\$K0F	TEST FOR F
1160	FCE7	CSAS 016		019	ISPLUS	BR IF DYN BIT7=1	TEST FOR PLUS
1162	C5F5	CSAS 017		026	ISCOMP	BR IF S0=1	TEST FOR COMP ADD
1164	3042	CSAS 018				SET S K=94	SET S0,S3,S5
1166	C07C	CSAS 019	ISPLUS	030	ENDSGN	BR IF GO BIT4=0	SIGN TEST IS OVER
1168	D07B	CSAS 020		029	ZAPFST	BR IF GO BIT5=1	BR IF ZERO AND ADD
116A	148B	CSAS 021				GO=GO\$K08	INVERT G04
116C	57FB	CSAS 022				H1=D1H	STRIP SIGN
116E	5710	CSAS 023				RDB D1 U+0	READ 2ND OPERAND BYTE
1170	9154	CSAS 024		010	TSTSGN	BR	
1172	8C7C	CSAS 025	NOSIGN	CCOM	071	FLGDEX	
1174	1002	CSAS 026	ISCOMP			RST S K=90	SET UP FOR TRUE ADD
1176	2040	CSAS 027				SET S5	
1178	9166	CSAS 028		019	ISPLUS	BR	
117A	1002	CSAS 029	ZAPFST			RST S K=90	RESET S0,S3
117C	577B	CSAS 030	ENDSGN			D1=D1H	STRIP SIGN
117E	77FF	CSAS 031				D1C=D1\$H1+C	ADD FIRST CHAR
1180	C905	CSAS 032		034	ISASCI	BR IF P1 BIT4=1	BR IF ASCII
1182	2728	CSAS 033				D1=D1+K02	
1184	D188	CSAS 034	ISASCI	036	DSPLUS	BR IF S5=0	BR IF PLUS
1186	271B	CSAS 035				D1=D1+K01	
1188	27AB	CSAS 036	DSPLUS			D1=D1+K0A	
118A	F035	CSAS 037		058	DECCMP	BR IF GO BIT7=1	TEST FOR COMPARE
118C	771A	CSAS 038				STB D1 U-1	
118E	55B9	CSAS 039	TSTLGT			T1=G1	
1190	E0EB	CSAS 040		085	DESTND	BR IF HZ=0	TEST FOR DEST END
1192	FOA6	CSAS 041		051	MORSOR	BR IF LZNZ	TEST FOR SOURCE LEFT
1194	27C7	CSAS 042	ENDSOR			D1=0	
1196	2EF3	CSAS 043				HO=0\$K0F	
1198	D02A	CSAS 044		053	RDEST	BR IF GO BIT5=0	
119A	2FC7	CSAS 045	ZRODST			H1=0	
119C	91AC	CSAS 046		054	COMPUT	BR	
119E	7F1A	CSAS 047	STORED			STB H1 U-1	
11A0	75E3	CSAS 048	DECLGT			G1=G1-H0	
11A2	E0EB	CSAS 049		085	DESTND	BR IF HZ=0	
11A4	F095	CSAS 050		042	ENDSOR	BR IF LZ=0	
11A6	573A	CSAS 051	MORSOR			RDB D1 V-1	
11A8	D01B	CSAS 052		045	ZRODST	BR IF GO BIT5=1	
11AA	5F10	CSAS 053	RDEST			RDB H1 U+0	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
11AC	7F7F	CSAS 054	COMPUT			HIC=H1@D1+C	
11AE	F01E	CSAS 055		047	STORED	BR IF GO BIT7=0	
11B0	50C6	CSAS 056				U=U-1	
11B2	91A0	CSAS 057		048	DECLGT	BR	
11B4	5006	CSAS 058	DECCMP			U=U-1	
11B6	918E	CSAS 059		039	TSTLGT	BR	
11B8	1413	CSAS 060	SSOPFB			GO=GO*-K01	SUBTRACT RST G07
11BA	3002	CSAS 061	SSOPF9			SET S K=90	COMPARE SET COMP&S3
11BC	9150	CSAS 062		008	STUPDC	BR	
11BE	573A	CSAS 063	COMPAD			RDB D1 V-1	
11C0	F5D1	CSAS 064		072	COMPGD	BR IF S3=1	
11C2	7A7F	CSAS 065				TOC=TO@D1+C	
11C4	1A9F	CSAS 066	TSTNIN			TO=TO@K99	
11C6	C4F9	CSAS 067		092	OKSOFA	BR IF Z=0	
11C8	00C2	CSAS 068				RST S3	
11CA	1415	CSAS 069	SETOVF			GO=GO*-K10	
11CC	2AC7	CSAS 070				TO=0	
11CE	91F8	CSAS 071		092	OKSOFA	BR	
11D0	E5D9	CSAS 072	COMPGD	076	MYBEOK	BR IF S2=1	
11D2	7A7F	CSAS 073				TOC=TO@D1+C	
11D4	C4F9	CSAS 074		092	OKSOFA	BR IF Z=0	
11D6	91CA	CSAS 075		069	SETOVF	BR	
11D8	7A7F	CSAS 076	MYBEOK			TOC=TO@D1+C	
11DA	F5E1	CSAS 077		080	STLCKA	BR IF S3=1	
11DC	E445	CSAS 078		066	TSTNIN	BR IF GO BIT2=1	
11DE	91CA	CSAS 079		069	SETOVF	BR	
11E0	142D	CSAS 080	STLOKA			GO=GO@K20	
11E2	91F8	CSAS 081		092	OKSOFA	BR	
11E4	77AF	CSAS 082	TRUCAR			DIC=D1@T0+C	
11E6	91F6	CSAS 083		091	OVFLAA	BR	
11E8	9172	CSAS 084	NOOSIN	025	NOSIGN	BR	
11EA	F0FD	CSAS 085	DESTND	094	ALLEND	BR IF LZ=0	
11EC	C5BF	CSAS 086	MORDST	063	COMPAD	BR IF S0=1	
11EE	573A	CSAS 087				RDB D1 V-1	
11F0	F5E5	CSAS 088		082	TRUCAR	BR IF S3=1	
11F2	77AF	CSAS 089				DIC=D1@T0+C	
11F4	C4F9	CSAS 090		092	OKSOFA	BR IF Z=0	
11F6	2002	CSAS 091	OVFLAA			SET S3	
11F8	25FF	CSAS 092	OKSOFA			G1=G1+KFF	
11FA	FOEC	CSAS 093		086	MORDST	BR IF LZNZ	
11FC	C1E9	CSAS 094	ALLEND	084	NOOSIN	BR IF S4=1	
11FE	C591	CSAS 095	ALENDH	104	ALENDH	BR IF S0=1	
1200	F589	CSAS 096		100	ALENCA	BR IF S3=1	
1202	E58B	CSAS 097	ALENDE	101	ALENDG	BR IF S2=1	
1204	D1D1	CSAS 098		136	ALENDI	BR IF S5=1	
1206	A8DC	CSAS 099	ALENDJ	CICY 003	CHECKX	BR	0
1208	F03C	CSAS 100	ALENDA	126	ALENDG	BR IF GO BIT7=0	
120A	D18F	CSAS 101	ALENDG	103	ALENDH	BR IF S5=1	
120C	ACA4	CSAS 102		CLOG 052	OP95A	BR	2
120E	AC84	CSAS 103	ALENDH	CLOG 050	OP95C	BR	1
1210	F583	CSAS 104	ALENDI	097	ALENDE	BR IF S3=1	
1212	F047	CSAS 105	ALENDH	131	ALENDI	BR IF GO BIT7=1	
1214	58B5	CSAS 106	RECOMP			T1=T1XL	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1216	61B9	CSAS 107				UIC=U1+T1+1	
1218	6CAD	CSAS 108				UOC=UO+TO+C	
121A	5710	CSAS 109				RDB D1 U+0	
121C	57FB	CSAS 110				H1=D1H	
121E	3002	CSAS 111				SET S K=90	
1220	577D	CSAS 112				D1=D1L	
1222	171B	CSAS 113				D1=D1K01	
1224	7AFF	CSAS 114				TOC=TO@H1+C	
1226	67A3	CSAS 115				D1=D1+TO	
1228	771A	CSAS 116				STB D1 U-1	
122A	58A9	CSAS 117				TO=T1	
122C	C4BB	CSAS 118		125	RCMPOV	BR IF Z=0	
122E	5710	CSAS 119	NXTCHR			RDB D1 U+0	
1230	757F	CSAS 120				G1C=G1@D1+C	
1232	751A	CSAS 121				STB G1 U-1	
1234	25C7	CSAS 122				G1=0	
1236	2AFF	CSAS 123				TO=TO+KFF	
1238	C4AE	CSAS 124		119	NXTCHR	BR IF ZNZ	
123A	F447	CSAS 125	RCMPOV	131	ALENDF	BR IF G03=1	
123C	2C75	CSAS 126	ALENDG			PO=0\$K70	SET CC 3
123E	DD43	CSAS 127		129	DEOVFL	BR IF P1 BIT1=1	
1240	A8DE	CSAS 128		CICY 004	CHECK	BR	
1242	2BA3	CSAS 129	DEOVFL			T1=0\$K0A	DECIMAL OVERFLOW
1244	9C70	CSAS 130		BPSW 003	PROG	BR	
1246	D1CD	CSAS 131	ALENDF	134	INVSNB	BR IF S5=1	
1248	2C40	CSAS 132				SET S5	
124A	9202	CSAS 133		097	ALENDE	BR	
124C	0040	CSAS 134	INVSNB			RST S5	
124E	9202	CSAS 135		097	ALENDE	BR	
1250	F007	CSAS 136	ALENDI	099	ALENDJ	BR IF G0 BIT7=1	
1252	9214	CSAS 137		106	RECCMP	BR	
		CSAS 138	AEND				

 * CROSS REFERENCE FOR CSECT CSAS *

CSAS 006	CICY 249		
CSAS 007	CICY 251		
CSAS 008	CSAS 062		
CSAS 010	CSAS 024		
CSAS 019	CSAS 014	CSAS 016	CSAS 028
CSAS 025	CSAS 012	CSAS 084	
CSAS 026	CSAS 017		
CSAS 029	CSAS 020		
CSAS 030	CSAS 019		
CSAS 034	CSAS 032		
CSAS 036	CSAS 034		
CSAS 039	CSAS 059		
CSAS 042	CSAS 050		
CSAS 045	CSAS 052		
CSAS 047	CSAS 055		
CSAS 048	CSAS 057		
CSAS 051	CSAS 041		
CSAS 053	CSAS 044		

* CROSS REFERENCE FOR CSECT CSAS *

CSAS 054	CSAS 046	
CSAS 058	CSAS 037	
CSAS 060	CICY 252	
CSAS 061	CICY 250	
CSAS 063	CSAS 086	
CSAS 066	CSAS 078	
CSAS 069	CSAS 075	CSAS 079
CSAS 072	CSAS 064	
CSAS 076	CSAS 072	
CSAS 080	CSAS 077	
CSAS 082	CSAS 088	
CSAS 084	CSAS 094	
CSAS 085	CSAS 040	CSAS 049
CSAS 086	CSAS 093	
CSAS 091	CSAS 083	
CSAS 092	CSAS 067	CSAS 071 CSAS 074 CSAS 081 CSAS 090
CSAS 094	CSAS 085	
CSAS 097	CSAS 104	CSAS 133 CSAS 135
CSAS 099	CSAS 136	
CSAS 100	CSAS 096	
CSAS 101	CSAS 097	
CSAS 103	CSAS 101	
CSAS 104	CSAS 095	
CSAS 106	CSAS 137	
CSAS 119	CSAS 124	
CSAS 125	CSAS 118	
CSAS 126	CSAS 100	
CSAS 129	CSAS 127	
CSAS 131	CSAS 105	CSAS 125
CSAS 134	CSAS 131	
CSAS 136	CSAS 098	

CSFT -- SHIFT ROUTINES

ENTRIES

1. RSOP89 -- SHIFT LEFT LOGICAL
- RSOP8B -- SHIFT LEFT ALGEBRAIC
- RSOP8D -- SHIFT LEFT DOUBLE LOGICAL
- RSOP8F -- SHIFT LEFT DOUBLE ALGEBRAIC
- RSOP88 -- SHIFT RIGHT LOGICAL
- RSOP8A -- SHIFT RIGHT ALGEBRAIC
- RSOP8C -- SHIFT RIGHT DOUBLE LOGICAL
- RSOP8E -- SHIFT RIGHT DOUBLE ALGEBRAIC

OBJECTIVES -- ALL SHIFTS

OBTAIN BINARY VALUE OF THE 2ND OPERAND EFFECTIVE ADDRESS LOW-ORDER 6 BITS. THEN SHIFT THE 1ST OPERAND EITHER LEFT OR RIGHT THAT AMOUNT.

DESCRIPTION

THE OPERANDS ARE FETCHED, THEN THE OP CODE IS TESTED. FOR ALGEBRAIC SHIFTS WITH NEGATIVE OPERAND, A REGISTER IS SET TO FF FOR EXTENDING THE SIGN. THE HIGH-ORDER BITS OF THE SHIFT-COUNT FIELD ARE HANDLED FIRST. THESE BITS REPRESENT SHIFTS OF 32, 16, AND 8. THE -SHIFT8- SUBROUTINE IS EXECUTED THE NUMBER OF TIMES NEEDED TO ACCOMPLISH ANY OF THE HIGH-ORDER SHIFTS.

AFTER THE HIGH-ORDER SHIFTS ARE COMPLETED, THE OP CODE IS TESTED FOR RIGHT OR LEFT SHIFT. THEN THE SHIFTS FOR THE LOW-ORDER 3 BITS OF THE COUNT FIELD ARE HANDLED. THESE BITS REPRESENT SHIFTS OF 4, 2, AND 1.

LEFT SHIFT 1 THROUGH 7. THE -LEFT1- SUBROUTINE IS USED. THIS SUBROUTINE SHIFTS 1 BIT, THEN DECREMENTS THE LOW-ORDER COUNT. THE SUBROUTINE LOOPS UNTIL THE COUNT IS REDUCED TO ZERO.

RIGHT SHIFT 1 THROUGH 7. IF A RIGHT SHIFT OF 4 IS NEEDED, IT IS HANDLED BY THE --RIGHT4- SUBROUTINE. THEN THE BITS REPRESENTING 8 AND 4 ARE RESET IN THE COUNT. THE BITS REPRESENTING 2 AND 1, FOR SHIFTS OF 0 THROUGH 3 ARE HANDLED AS FOLLOWS.

LOW 2 BITS OF COUNT

- | | |
|--------------------|--------------------------------------|
| 00 DONE. | GO TO SHIFT COMPLETE |
| 01 SET COUNT TO 3. | RIGHT 1 EQUALS 3 LEFT MINUS 4 RIGHT. |
| 10 COUNT OKAY. | RIGHT 2 EQUALS 2 LEFT MINUS 4 RIGHT. |
| 11 SET COUNT TO 1. | RIGHT 3 EQUALS 1 LEFT MINUS 4 RIGHT. |

DESCRIPTION CONTINUED

AFTER THE COUNT IS SET UP, THE -LEFT1- SUBROUTINE LOOPS UNTIL THE COUNT IS REDUCED TO ZERO. THEN THE RIGHT SHIFT OF 4 IS DONE BY THE -RIGHT4- SUBROUTINE.

SHIFT COMPLETE. THIS SECTION OF THE ROUTINE STORES THE RESULT BACK INTO THE 1ST OPERAND. FOR LOGICAL SHIFTS, THE MICROPROGRAM RESTORES SAVED REGISTERS AND RETURNS TO I-CYCLES. FOR ALGEBRAIC SHIFTS, SO IS TESTED TO DETERMINE IF AN OVERFLOW OCCURRED. FOR SHIFT OVERFLOW, THE SIGN IS INVERTED AND STORED, AND THE MICROPROGRAM EXITS TO SET CONDITION CODE 3. IF NO OVERFLOW, THE MICROPROGRAM SETS THE CONDITION CODE ACCORDING TO THE SIGN AND WHETHER THE RESULT IS ZERO OR NON-ZERO. THE SAVED REGISTERS ARE RESTORED, AND THE MICROPROGRAM RETURNS TO I-CYCLES.

SUBROUTINES

-SHIFT8-. THIS SUBROUTINE HANDLES BOTH RIGHT AND LEFT SHIFTS. SHIFT RIGHT 8 IS ACCOMPLISHED BY A STRAIGHT MOVE OF EACH BYTE-SOURCE REGISTER TO THE NEXT ADJACENT REGISTER ON THE RIGHT. VACATED HIGH-ORDER BITS ARE FILLED FROM A REGISTER THAT CONTAINS BITS LIKE THE ORIGINAL SIGN.

SHIFT LEFT 8 IS ACCOMPLISHED BY A STRAIGHT MOVE OF EACH BYTE-SOURCE REGISTER TO THE NEXT ADJACENT REGISTER ON THE LEFT. ANY CHANGE IN SIGN IS AN OVERFLOW AND MUST BE REMEMBERED TO SET CONDITION CODE 3 FOR ALGEBRAIC SHIFTS.

-LEFT1-. SHIFT LEFT 1 IS ACCOMPLISHED BY ADDING EACH BYTE TO ITSELF, SAVING CARRIES. SIGN CHANGE IS STORED BY SETTING SO TO INDICATE OVERFLOW.

-RIGHT4-. EACH HALF BYTE IS CROSSED INTO THE NEXT ADJACENT HALF-BYTE ON THE RIGHT.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CSFT 001	T			SHIFT ROUTINES	
		CSFT 002	*				
		CSFT 003	*				
		CSFT 004	*				
		CSFT 005	*			ALL 8 SHIFTS START AT THE FIRST WORD OF THIS ROUTINE	
		CSFT 006	*			DOUBLE SHIFT DATA IS IN H,D,V,U	
		CSFT 007	*			SINGLE SHIFT DATA IS IN H,D	
		CSFT 008	*			ORIGINAL SIGN EXPANDED TO 8 BITS =TO	
		CSFT 009	*			COUNT FOR LOOPS IS IN PO OR P1	
		CSFT 010	*			NO OF BIT POSITIONS TO BE SHIFTED IS IN P1	
		CSFT 011	*			S5=ORIG SIGN- , SO USED FOR OVERFLOW	
		CSFT 012	*			*****	
		CSFT 013	*				
18F4	10CE	CSFT 014	SHIFT			RST S K=FC	
18F6	7CC2	CSFT 015				STH P DA,B8	
18F8	51D9	CSFT 016				P1=U1	SAVE SHIFT COUNT
18FA	55BB	CSFT 017				T1=G1H	GET R1
18FC	5EA8	CSFT 018				RDH H AS,T+2	
18FE	56AA	CSFT 019				RDH D AS,T-2	
1C00	00CE	CSFT 020		027	START	BR IF G05=0	BR IF SHORT
1C02	F508	CSFT 021		024	NOSP	BR IF G13=0	BR IF R1 EVEN
1C04	5CC2	CSFT 022				RDH P DA,B8	RESTORE P
1C06	AE98	CSFT 023		CICY 059	SPECHK	BR	
1C08	3B15	CSFT 024	NOSP			T1=T1\$K10	GET SECOND HALF
1C0A	52A8	CSFT 025				RDH V AS,T+2	OF
1C0C	50AA	CSFT 026				RDH U AS,T-2	DOUBLE OPERAND
1C0E	E016	CSFT 027	START	031	LOGPOS	BR IF G06=0	BR IF LOGICAL OP
1C10	CE16	CSFT 028		031	LOGPOS	BR IF H00=0	BR IF +
1C12	2AF7	CSFT 029				T0=0\$KFF	EXTEND - SIGN
1C14	2040	CSFT 030				SET S5	REMEMBER NEG
1C16	5DC9	CSFT 031	LOGPOS			PO=P1	MOVE SHIFT COUNT
1C18	6CC3	CSFT 032				PO=PO+PO	MOVE THE 32,16,8 CTS LEFT ONE
1C1A	1C85	CSFT 033				PO=PO*-K80	
1C1C	E0A5	CSFT 034		038	NOB	BR IF HZ=0	BR IF NO 32,16 OR 8 SHIFT
1C1E	B29A	CSFT 035	BAL	129	SHIFT8	BAL	SHIFT ONE BYTE LEFT OR RIGHT
1C20	2CFD	CSFT 036				PO=PO+KFO	DEC HI COUNT
1C22	E09E	CSFT 037		035	BAL	BR IF HZNZ	
1C24	F02F	CSFT 038	NOB	054	LEFT	BR IF G07=1	BREAK OUT LEFT SHIFTS
1C26	D92A	CSFT 039		041	NO4	BR IF P15=0	BR IF NO RIGHT 4
1C28	B04C	CSFT 040		107	RIGHT4	BAL	
1C2A	1DC3	CSFT 041	NO4			P1=P1*-K0C	
1C2C	8975	CSFT 042		046	LOW N	N=P1 BITS67	
		CSFT 043	*			*****	
		CSFT 044	*		RIGHT 1=3L-4R	** RIGHT 2=2L-4R ** RIGHT 3=1L-4R	
		CSFT 045	*			LOW ORDER BITS MUST BE CORRECTED	
1AE0	9C34	CSFT 046	LOW 0	057	SHFTCC	BR	NO 1,2 OR 3 RIGHT SHIFT
1AE2	2D2B	CSFT 047	LOW 1			P1=P1+K02	R1=3L-4R FIX P1=3
1AE4	9AE8	CSFT 048	LOW 2	050	BALRL	BR	R2=2L-4R P1=OK
1AE6	1D23	CSFT 049	LOW 3			P1=P1*-K02	R3=1L-4R FIX P1
1AE8	A460	CSFT 050	BALRL	087	LEFT1	BAL	
1AEA	B04C	CSFT 051		107	RIGHT4	BAL	
1AEC	9C34	CSFT 052		057	SHFTCC	BR	
		CSFT 053	*			*****	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1C2E	1D83	CSFT 054	LEFT			P1=P1*-K08	REMOVE HI BIT
1C30	F0B5	CSFT 055		057	SHFTCC	BR IF LZ=0	
1C32	A460	CSFT 056		087	LEFT1	BAL	DO LEFT SHIFT 1 THRU 7 PLACES
1C34	2AC7	CSFT 057	SHFTCC			TO=0	REMOVE EXT SIGN
1C36	5CC2	CSFT 058				RDH P DA, B8	RESTORE MASK
1C38	D040	CSFT 059		063	SHST	BR IF G05=0	
1C3A	72A8	CSFT 060				STH V AS, T+2	STORE
1C3C	70AA	CSFT 061				STH U AS, T-2	LOW
1C3E	1B15	CSFT 062				T1=T1*-K10	DOUBLE
1C40	7EA8	CSFT 063	SHST			STH H AS, T+2	STORE
1C42	76AA	CSFT 064				STH D AS, T-2	SINGLE OR HI DOUBLE
1C44	E049	CSFT 065		067	ALG	BR IF G06=1	
1C46	A8DE	CSFT 066		CICY 004	CHECK	BR	LOGICAL GO BACK
1C48	F058	CSFT 067	ALG	075	NOINT	BR IF G07=0	BR IF RIGHT-NO OVF CHECK
1C4A	C5D8	CSFT 068		075	NOINT	BR IF S0=0	LEFT ALG CHECK OVF
1C4C	56A0	CSFT 069				RDH D AS, T+0	HAVE OVF FIX SIGN
1C4E	1685	CSFT 070				DO=DO*-K80	MAKE SIGN +
1C50	D1D4	CSFT 071		073	CSIGN	BR IF S5=0	BR IF ORIGINAL SIGN +
1C52	3685	CSFT 072				DO=DO\$K80	MAKE SIGN -
1C54	76A0	CSFT 073	CSIGN			STH D AS, T+0	STOR CORRECT SIGN
1C56	A840	CSFT 074		CLST 011	SHFTOV	BR	
1C58	CE5E	CSFT 075	NOINT	077	POS	BR IF H00=0	
1C5A	A91A	CSFT 076		CLOG 082	SETNZ	BR	CC 1 -
1C5E	0008	CSFT 077	POS			RST S1	
1C60	4EE6	CSFT 078				H=H	TEST
1C62	4666	CSFT 079				D=D	HI
1C64	D06A	CSFT 080		083	SHCRT	BR IF G05=0	BR IF SINGLE
1C66	4226	CSFT 081				V=V	TEST
1C68	40C6	CSFT 082				U=U	LOW DOUBLE
1C6A	D5DD	CSFT 083	SHORT	085	CC2	BR IF S1=1	
1C6C	A8DC	CSFT 084		CICY 003	CHECKX	BR	CCC
1C6E	ACA4	CSFT 085	CC2	CLOG 052	OP95A	BR	
1C5C		CSFT 086	*			*** LEFT 1 ROUTINE	***
2460	0002	CSFT 087	LEFT1			RST S3	
2462	D06C	CSFT 088		093	SHORT1	BR IF G05=0	
2464	611B	CSFT 089				U1C=U1+U1	SHIFT
2466	600D	CSFT 090				U0C=U0+U0+C	
2468	633D	CSFT 091				V1C=V1+V1+C	LEFT
246A	622D	CSFT 092				V0C=V0+V0+C	
246C	677D	CSFT 093	SHORT1			D1C=D1+D1+C	ONE
246E	666D	CSFT 094				D0C=D0+D0+C	
2470	6FFD	CSFT 095				H1C=H1+H1+C	BIT
2472	6EED	CSFT 096				H0C=H0+H0+C	
2474	E4F8	CSFT 097		099	SKIPST	BR IF NOVFL	
2476	3000	CSFT 098				SET S0	
2478	6AAD	CSFT 099	SKIPST			TOC=TO+TO+C	SAVE CARRY FOR RIGHT SHIFTS
247A	2DFB	CSFT 100				P1=P1+K0F	DEC COUNT
247C	F0E0	CSFT 101		087	LEFT1	BR IF LZNZ	
247E	128E	CSFT 102				RTN	
		CSFT 103	*				
		CSFT 104	*				
		CSFT 105	*				
		CSFT 106	*				

*** RIGHT 4 ROUTINE ***

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
304C	D05E	CSFT 107	RIGHT4	116	R4SHOR	BR IF G05=0	
304E	5115	CSFT 108				U1=U1XL	
3050	4013	CSFT 109				U1=U0XH+U1L	
3052	5005	CSFT 110				U0=U0XL	
3054	4303	CSFT 111				U0=V1XH+U0L	
3056	5335	CSFT 112				V1=V1XL	
3058	4233	CSFT 113				V1=V0XH+V1L	
305A	5225	CSFT 114				V0=V0XL	
305C	4723	CSFT 115				V0=D1XH+V0L	
305E	5775	CSFT 116	R4SHOR			D1=D1XL	
3060	4673	CSFT 117				D1=D0XH+D1L	
3062	5665	CSFT 118				D0=D0XL	
3064	4F63	CSFT 119				D0=H1XH+D0L	
3066	5FF5	CSFT 120				H1=H1XL	
3068	4EF3	CSFT 121				H1=H0XH+H1L	
306A	5EE5	CSFT 122				H0=H0XL	
306C	4AE3	CSFT 123				H0=TOXH+H0L	
306E	128E	CSFT 124				RTN	
		CSFT 125	*				
		CSFT 126	*				
		CSFT 127	*				
		CSFT 128	*				
						** COMMON SHIFT 8	RIGHT AND LEFT ***
329A	F001	CSFT 129	SHIFT8	140	LEFT8	BR IF G07=1	BR IF LEFT SHIFT
329C	D026	CSFT 130		135	SHORTR	BR IF G05=0	THIS IS RIGHT8
329E	5019	CSFT 131				U1=U0	SHIFT
32A0	5309	CSFT 132				U0=V1	
32A2	5239	CSFT 133				V1=V0	RIGHT
32A4	5729	CSFT 134				V0=D1	
32A6	5679	CSFT 135	SHORTR			D1=D0	ONE
32A8	5F69	CSFT 136				D0=H1	
32AA	5EF9	CSFT 137				H1=H0	BYTE
32AC	5AE9	CSFT 138				H0=TO	SHIFT
32AE	128E	CSFT 139				RTN	
3280	6EA1	CSFT 140	LEFT8			H0=H0TO	CHECK HI 8 BITS FOR SIGN CHANGE
3282	C48C	CSFT 141		146	OVFSET	BR IF ZNZ	
3284	D18A	CSFT 142		145	PLUS	BR IF S5=0	
3286	CF0F	CSFT 143		147	OK	BR IF H10=1	ORIG SIGN-BR IF NEW SIGN-
3288	3000	CSFT 144				SET S0	ORIG SIGN-NEW SIGN+REMBER OVF
328A	CF0E	CSFT 145	PLUS	147	OK	BR IF H10=0	BR IF NEW SIGN BIT +
328C	3000	CSFT 146	OVFSET			SET S0	REMEMBER OVERFLOW
328E	5FE9	CSFT 147	OK			H0=H1	SHIFT
3290	56F9	CSFT 148				H1=D0	
3292	5769	CSFT 149				D0=D1	LEFT
3294	DC31	CSFT 150		153	LONGL8	BR IF G05=1	
3296	2707	CSFT 151				D1=0	ONE
3298	128E	CSFT 152				RTN	
32B0	5279	CSFT 153	LONGL8			D1=V0	BYTE
32B2	5329	CSFT 154				V0=V1	
32B4	5039	CSFT 155				V1=U0	
32B6	5109	CSFT 156				U0=U1	
32B8	2107	CSFT 157				U1=0	
32BA	128E	CSFT 158				RTN	

* CROSS REFERENCE FOR CSECT CSFT *

CSFT 014	CICY 050		
CSFT 024	CSFT 021		
CSFT 027	CSFT 020		
CSFT 031	CSFT 027	CSFT 028	
CSFT 035	CSFT 037		
CSFT 038	CSFT 034		
CSFT 041	CSFT 039		
CSFT 046	CSFT 042		
CSFT 050	CSFT 048		
CSFT 054	CSFT 038		
CSFT 057	CSFT 046	CSFT 052	CSFT 055
CSFT 063	CSFT 059		
CSFT 067	CSFT 065		
CSFT 073	CSFT 071		
CSFT 075	CSFT 067	CSFT 068	
CSFT 077	CSFT 075		
CSFT 083	CSFT 080		
CSFT 085	CSFT 083		
CSFT 087	CSFT 050	CSFT 056	CSFT 101
CSFT 093	CSFT 088		
CSFT 099	CSFT 097		
CSFT 107	CSFT 040	CSFT 051	
CSFT 116	CSFT 107		
CSFT 129	CSFT 035		
CSFT 135	CSFT 130		
CSFT 140	CSFT 129		
CSFT 145	CSFT 142		
CSFT 146	CSFT 141		
CSFT 147	CSFT 143	CSFT 145	
CSFT 153	CSFT 150		

CSTS -- STATUS SWITCHING INSTRUCTIONS

ENTRIES

RSOP82 -- LOAD PSW	RSOP93 -- TEST AND SET
RSOP80 -- SET SYSTEM MASK	RROPO4 -- SET PROGRAM MASK
RSOP83 -- DIAGNOSE	RROPOA -- SUPERVISOR CALL
RROPO8 -- SET STORAGE KEY	RSOP84 -- WRITE DIRECT
RROPO8 -- INSERT STORAGE KEY	RSOP85 -- READ DIRECT

OBJECTIVES

GENERAL. THE FOLLOWING TESTS ARE MADE IF APPLICABLE TO INSTRUCTION BEING EXECUTED.

1. CHECK THAT ADDRESS DOES NOT EXCEED 16 BITS.
2. IF PRIVILEGED OP, TEST FOR SUPERVISOR STATE.
3. CHECK ADDRESS BOUNDARY.

LOAD PSW

THE DOUBLE-WORD AT THE LOCATION DESIGNATED BY THE EFFECTIVE OPERAND ADDRESS REPLACES THE PSW.

THE MICROROUTINE TESTS FOR ERRORS, THEN BRANCHES TO THE -BPSW- ROUTINE TO LOAD THE PSW.

SET SYSTEM MASK

THE BYTE AT THE 2ND OPERAND EFFECTIVE ADDRESS REPLACES THE SYSTEM MASK, BITS 0-7, OF THE CURRENT PSW.

AFTER TESTING FOR ERRORS, THE ROUTINE READS THE 2ND OPERAND. THE NEW MASK BYTE IS PLACED IN THE SYSTEM MASK REGISTER -SM- AND IN AUXILIARY STORAGE LOCATION 00A8.

DIAGNOSE

THE DIAGNOSE INSTRUCTION PROVIDES A MEANS OF GOING DIRECTLY TO A SPECIFIC MICROINSTRUCTION. THE ADDRESS OF THIS MICROINSTRUCTION IS GIVEN IN THE OPERAND OF THE DIAGNOSE INSTRUCTION.

THE OPERAND EFFECTIVE ADDRESS IS STORED IN AUXILIARY STORAGE LOCATION 00BC. THEN THIS ADDRESS IS READ FROM AUXILIARY STORAGE INTO THE BACKUP-ZONE I-REG. AFTER THE L.S. ZONE IS SET BACK TO 0, A RETURN WORD IS EXECUTED. THIS CAUSES THE BACKUP I-REG TO BE USED TO ADDRESS THE NEXT MICROINSTRUCTION.

SET STORAGE KEY

SET THE STORAGE-PROTECT KEY FOR THE PROGRAM-STORAGE BLOCK ADDRESSED BY R2.

THE ROUTINE READS THE BLOCK ADDRESS AND KEY. NEXT THE KEY IS PUT IN THE Q-REG USING STPO MNEMONIC. THEN, THE KEY IS STORED IN STP1 AT THE SPECIFIED BLOCK ADDRESS. BEFORE RETURNING TO I-CYCLES, THE CPU KEY IS RESTORED.

INSERT STORAGE KEY

READ THE STORAGE-PROTECT KEY FOR THE PROGRAM-STORAGE BLOCK ADDRESSED BY R2, AND SET IT IN R1.

THE ROUTINE READS THE BLOCK ADDRESS FROM R2. NEXT, THE KEY IS READ FROM STP1 INTO THE Q-REG. THE KEY IS PUT INTO R1, THEN THE CPU KEY IS RESTORED.

TEST AND SET

TEST THE HIGH-ORDER BIT OF THE BYTE AT THE EFFECTIVE ADDRESS, AND SET THE CONDITION CODE.

THE ROUTINE READS THE ADDRESSED BYTE, THEN TESTS THE HIGH-ORDER BIT FOR 1. IF THE BIT IS 1, CONDITION CODE 1 IS SET. IF THE BIT IS 0, CONDITION CODE 0 IS SET.

SET PROGRAM MASK

BITS 2-7 OF R1 REPLACE THE CONDITION CODE AND THE PROGRAM MASK BITS OF THE CURRENT PSW.

THE CONDITION CODE IS SET IN P0 AND THE PROGRAM MASK IS SET IN P1.

SUPERVISOR CALL

THIS INSTRUCTION CAUSES A SUPERVISOR-CALL INTERRUPTION. THE INTERRUPT CODE IS SUPPLIED IN BYTE 2 OF THE INSTRUCTION.

THE MICROPROGRAM SETS UP A POINTER TO THE SVC OLD PSW, THEN BRANCHES TO THE LOAD PSW ROUTINE.

WRITE DIRECT

THE BYTE AT THE LOCATION DESIGNATED BY THE OPERAND ADDRESS IS MADE AVAILABLE AS A SET OF DIRECT-OUT STATIC SIGNALS. INSTRUCTION BITS 8-15 ARE MADE AVAILABLE AS SIGNAL-OUT TIMING SIGNALS.

READ DIRECT

A DIRECT-IN DATA BYTE IS ACCEPTED FROM AN EXTERNAL DEVICE IN THE ABSENCE OF A HOLD SIGNAL AND IS PLACED IN THE LOCATION DESIGNATED BY THE OPERAND ADDRESS. INSTRUCTION-BITS 8-15 ARE MADE AVAILABLE AS SIGNAL-OUT TIMING SIGNALS.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CSTS 001	T			STATUS SWITCHING INSTRUCTIONS	
		CSTS 002	*				
2B7E	AB53	CSTS 003	RSOP82	CICY 045	ADDERR N	BR IF H1=NZ	LOAD PSW
2B80	F90D	CSTS 004		020	OP80A	BR IF P17=1	OPERATION
2B82	0183	CSTS 005				Z=U1*-K08	CODE
2B84	F088	CSTS 006		009	OP82A	BR IF LZNZ	EXECUTION
2B86	9C9A	CSTS 007		BPSW 024	LPSW	BR	
		CSTS 008	*				
2B88	AE98	CSTS 009	OP82A	CICY 059	SPECHK	BR	
		CSTS 010	*			SET SYSTEM MASK	
2BC2	AB53	CSTS 011	RSOP80	CICY 045	ADDERR N	BR IF H1=NZ	SET SYSTEM
2BC4	F90D	CSTS 012		020	OP80A	BR IF P17=1	MASK
2BC6	5710	CSTS 013				RDB D1 U	OP
2BC8	447F	CSTS 014				SM=D1	CODE
2BCA	7782	CSTS 015				STB D1 DA,A8	EXECUTION
2BCC	81B3	CSTS 016	OP81A	CICY 005	ISTART N	N=S BITS67	
		CSTS 017	*				
		CSTS 018	*			DIAGNOSE INSTRUCTION	
2B8A	F938	CSTS 019	RSOP83	029	OP83A	BR IF P17=0	
2B8C	B35A	CSTS 020	OP80A	CICY 068	PRIVOP	BR	PRIV OP
2BB8	70E2	CSTS 029	OP83A			STH U DA,BC	DIAGNOSE
2BBA	2440	CSTS 031				SET MODE K=04	
2BBC	58E2	CSTS 032				RDH I DA,BC	
2BBE	2400	CSTS 033				SET MODE K=00	
2BC0	128E	CSTS 034				RTN	
		CSTS 035	*			RROPS 08-SET KEY AND 09 INSERT KEY	SHARE THIS ROUTINE
2B8E	F90D	CSTS 039	RROPO8	020	OP80A	BR IF P17=1	CHECK PRIV. OP
2B90	55B3	CSTS 040				T1=G1XH	PUT R2 ADDRESS IN T1
2B92	5EA8	CSTS 041				RDH H AS,T+2	
2B94	AB53	CSTS 042		CICY 045	ADDERR N	BR IF H1=NZ	
2B96	50AA	CSTS 043				RDH U AS,T-2	GET BLOCK ADDRESS
2B98	110D	CSTS 044				U1=U1#K00	
2B9A	F088	CSTS 045		009	OP82A	BR IF LZNZ	CHECK FOR SPEC ERROR
2B9C	55BB	CSTS 046				T1=G1H	GET R1
2B9E	3B33	CSTS 047				T1=T1#K03	T1=XXXX0011
2BA0	F02F	CSTS 048		055	INSERT	BR IF G07=1	BR IF INSERT KEY
2BA2	57A0	CSTS 049				RDB D1 AS,T+0	THIS IS SET KEY-KEY IS IN D1H
2BA4	17F3	CSTS 050				D1=D1*-K0F	D1=KKK0000
2BA6	427F	CSTS 051				STPO=D1	
2BA8	531C	CSTS 052				SSK STP1 U+1	
2BAA	A7FA	CSTS 053	DONE	CCDM 181	RSTRKY	BAL	RESTORE CPU KEY
2BAC	81B3	CSTS 054		CICY 005	ISTART N	N=S BITS67	
2BAE	521C	CSTS 055	INSERT			ISK STPO U+2	READ KEY
2BB0	527F	CSTS 056				D1=STPO	
2BB2	17F3	CSTS 057				D1=D1*-K0F	
2BB4	77A0	CSTS 058				STB D1 AS,T+0	STORE KEY IN BITS 24-27 OF R1
2BB6	ABAA	CSTS 059		053	DONE	BR	GO TO RESTORE CPU KEY
		CSTS 061	*				
		CSTS 062	ATABLE	ADDR=0AD4			
0AD4	8ADA	CSTS 063		066	RDSWCH	BAL	
0AD6	7098	CSTS 064				STH U I+2	
0AD8	A8DE	CSTS 065		CICY 004	CHECK	BR	
0ADA	500F	CSTS 066	RDSWCH			U0=SWAB	READ

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
OADC	511F	CSTS 067				U1=SWCD	SWITCHES
OADE	128E	CSTS 068				RTN	ROUTINE
		CSTS 069	AEND				
		CSTS 070	*				
						TEST AND SET	
2E6C	AB53	CSTS 071	RSOP93	CICY 045	ADDERR N	BR IF H1=NZ	TEST
2E6E	2FF7	CSTS 072				H1=0\$KFF	AND
2E70	5710	CSTS 073				RDB D1 U	SET
2E72	7F10	CSTS 074				STB H1 U	EXECUTION
2E74	C77A	CSTS 075		078	OP93A	BR IF D10=0	
2E76	2C55	CSTS 076				PO=0\$K50	
2E78	81B3	CSTS 077		CICY 005	ISTART N	N=S BITS67	
2E7A	2C05	CSTS 078	OP93A			PO=0\$K00	
2E7C	81B3	CSTS 079		CICY 005	ISTART N	N=S BITS67	
		CSTS 080	*				
		CSTS 081	*			SET PROGRAM MASK	
2E36	558B	CSTS 082	RROP04			T1=G1H	SET
2E38	5EA0	CSTS 083				RDH H AS,T	PROG
2E3A	4ED3	CSTS 084				P1=HOXH+P1L	MASK
2E3C	5ECB	CSTS 085				PO=HOH	EXECUTION
2E3E	1CC5	CSTS 086				PO=PO*-KCO	
2E40	FC44	CSTS 087		089	OP04A	BR IF P03=0	
2E42	3C45	CSTS 088				PO=PO\$K40	
2E44	7CC2	CSTS 089	OP04A			STH P DA,B8	
2E46	81B3	CSTS 090		CICY 005	ISTART N	N=S BITS67	
		CSTS 091	*				
		CSTS 092	*			SUPERVISOR CALL	
18F6	5589	CSTS 093	RROPOA			T1=G1	SUPERVISOR
18F8	2125	CSTS 094	SVC			U1=0\$K20	CALL
18FA	2FC7	CSTS 095	PICKUP			H1=0	EXECUTION
18FC	217B	CSTS 096				U1=U1+K07	
18FE	9C76	CSTS 097		BPSW 006	ENTRY	BR	
		CSTS 098	*				
		CSTS 099	*			* DIRECT CONTROL OPS	
3302	F901	CSTS 104	RSOP84	118	PRIVOP	BR IF P1 BIT7=1	WRITE DIREDT OPERATION
3304	5310	CSTS 105				RDB V1 U	GET DATA TO SEND
3306	495F	CSTS 106				JA=G1	SET SIGNAL OUT
3308	473F	CSTS 107				JD=V1	SEND DATA TO BUSS OUT
330A	A8DE	CSTS 108		CICY 004	CHECK	BR	RETURN TO NEXT INSTRUCTION
330C	F901	CSTS 109	RSOP85	118	PRIVOP	BR IF P1 BIT7=1	READ DIRECT OPRRATION
330E	495F	CSTS 110				JA=G1	SET SIGNAL OUT
3310	3F00	CSTS 111				SET CPF K=80	SET READ ENABLE
3312	CC52	CSTS 112	HOLDIN	112	HOLDIN	BR IF DYN BIT4=0	WAIT FOR HOLD IN TO FALL
3314	585F	CSTS 113				G1=J1	READ DATA FROM BUSS IN
3316	1F00	CSTS 114				RST CPF K=80	RESET READ ENABLE
3318	15FF	CSTS 115				G1=G1\$KFF	INVERT DATA
331A	7510	CSTS 116				STB G1 U	STORE DATA
331C	A8DE	CSTS 117		CICY 004	CHECK	BR	RETURN TO NEXT INSTRUCTINN
3300	B35A	CSTS 118	PRIVOP	CICY 068	PRIVOP	BR	GO TO PRIVALEGE OP INTERRUPT

* CROSS REFERENCE FOR CSECT CSTS *

CSTS 003	CICY 197		
CSTS 009	CSTS 006	CSTS 045	
CSTS 011	CICY 195		
CSTS 019	CICY 198		
CSTS 020	CSTS 004	CSTS 012	CSTS 039
CSTS 029	CSTS 019		
CSTS 039	CICY 130	CICY 131	
CSTS 053	CSTS 059		
CSTS 055	CSTS 048		
CSTS 066	CSTS 063	DCLL 015	DCLL 037
CSTS 071	CICY 206		
CSTS 078	CSTS 075		
CSTS 082	CICY 126		
CSTS 089	CSTS 087		
CSTS 093	CICY 132		
CSTS 095	BMCK 019		
CSTS 104	CICY 199		
CSTS 109	CICY 200		
CSTS 112	CSTS 112		
CSTS 118	CSTS 104	CSTS 109	

CTRKT -- TRANSLATE, TRANSLATE AND TEST, EDIT, EDIT AND MARK

ENTRIES

SSOPDC -- TRANSLATE, TRANSLATE AND TEST
 SSOPDE -- EDIT, EDIT AND MARK

OBJECTIVES

TRANSLATE

THE BYTES OF THE 1ST OPERAND ARE USED AS ARGUMENTS TO REFERENCE THE LIST DESIGNATED BY THE 2ND OPERAND ADDRESS. EACH FUNCTION BYTE SELECTED FROM THE LIST REPLACES THE CORRESPONDING ARGUMENT IN THE 1ST OPERAND.

THE BYTES OF THE 1ST OPERAND ARE SELECTED ONE BY ONE FOR TRANSLATION, PROCEEDING LEFT TO RIGHT, UNTIL THE 1ST OPERAND FIELD IS EXHAUSTED.

TRANSLATE AND TEST

THE BYTES OF THE 1ST OPERAND ARE USED AS ARGUMENTS TO REFERENCE THE LIST DESIGNATED BY THE 2ND OPERAND ADDRESS. EACH FUNCTION BYTE THUS SELECTED FROM THE LIST DETERMINES THE CONTINUATION OF THE OPERATION. WHEN THE FUNCTION BYTE IS ZERO, THE OPERATION PROCEEDS BY FETCHING AND TRANSLATING THE NEXT ARGUMENT BYTE. WHEN THE FUNCTION BYTE IS NONZERO, THE OPERATION IS COMPLETED BY INSERTING THE RELATED ARGUMENT ADDRESS IN THE LOW-ORDER 24 BITS OF GENERAL REGISTER 1, AND BY INSERTING THE FUNCTION BYTE IN THE LOW-ORDER BYTE OF GENERAL REGISTER 2.

EDIT

THE FORMAT OF THE SOURCE, 2ND OPERAND, IS CHANGED FROM PACKED TO ZONED AND IS MODIFIED UNDER CONTROL OF THE PATTERN IN THE 2ND OPERAND.

DURING THE EDITING PROCESS, EACH CHARACTER OF THE PATTERN IS AFFECTED AS FOLLOWS.

1. IT IS LEFT UNCHANGED.
2. IT IS REPLACED BY THE 1ST CHARACTER IN THE PATTERN CALLED THE FILL CHARACTER.
3. IT IS REPLACED BY A SOURCE DIGIT EXPANDED TO ZONED-FORMAT.

WHICH ACTION TAKES PLACE IS DETERMINED BY -- THE TYPE OF PATTERN CHARACTER, THE SIGNIFICANCE INDICATOR, AND WHETHER THE SOURCE DIGIT IS ZERO.

PATTERN CHARACTERS

- 0010 0000 DIGIT SELECTOR. FILL CHARACTER OR ZONED DIGIT REPLACES THE SOURCE DIGIT DEPENDING ON SOURCE DIGIT AND SIGNIFICANCE INDICATOR.
- 0010 0001 SIGNIFICANCE STARTER. SAME AS DIGIT SELECTOR, EXCEPT 0 CAN START SIGNIFICANCE.
- 0010 0010 FIELD SEPARATOR. FILL CHARACTER REPLACES SOURCE DIGIT, AND SIGNIFICANCE INDICATOR IS TURNED OFF.
- OTHER -- MESSAGE CHARACTER. THESE CHARACTERS REMAIN UNCHANGED OR ARE REPLACED BY THE FILL CHARACTER DEPENDING ON SIGNIFICANCE INDICATOR.

EDIT AND MARK

SAME AS EDIT. IN ADDITION, THE ADDRESS OF THE 1ST SIGNIFICANT RESULT DIGIT IS INSERTED IN BITS 8-31 OF GENERAL REGISTER 1.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CTRKT 001	T			TRANSLATE AND TRANSLATE AND TEST	C.V.PERKINS
		CTRKT 002	*				
2D7C	AB53	CTRKT 003	SSOPDC	CICY 045	ADDERR N	BR IF H1=NZ	BR IF INVALID ADDER
2D7E	2607	CTRKT 004				DO=0	
2D80	2E07	CTRKT 005	NXTCHR			HO=0	
2D82	5F10	CTRKT 006				RDB H1 U	READ ARGUMENT BYTE
2D84	6F3B	CTRKT 007				H1C=H1+V1	CALCULATE
2D86	6E2D	CTRKT 008				HO=C=HO+VO+C	SECOND
2D88	667D	CTRKT 009				DO=C=DO+D1+C	OPERAND

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2D8A	A253	CTRT 010		CICY 045	ADDERR N	BR IF D0=NZ	ADDERR ERROR IF BRANCH
2D8C	5FF0	CTRT 011				RDB H1 H	READ TABLE
2D8E	F019	CTRT 012		017	TRT	BR IF G07=1	BR IF TRT INSTR
2D90	7F18	CTRT 013				STB H1 U+1	
2D92	25FF	CTRT 014				G1=G1+KFF	DECREMENT COUNT
2D94	F481	CTRT 015		005	NXTCHR	BR IF AC=1	
2D96	81B3	CTRT 016		CICY 005	ISTART N	N=S BITS67	
2D98	AB13	CTRT 017	TRT	022	NONZRO N	BR IF H1=NZ	
2D9A	5004	CTRT 018				U=U+1	
2D9C	25FF	CTRT 019				G1=G1+KFF	
2D9E	F481	CTRT 020		005	NXTCHR	BR IF AC=1	
2DA0	A8DC	CTRT 021		CICY 003	CHECKX	BR	
2980	2B33	CTRT 022	NONZRO 0			T1=0\$K03	
2982	2B2D	CTRT 023				T1=T1+K20	
2984	7FA0	CTRT 024				STB H1 AS,T	
2986	2BFD	CTRT 025				T1=T1+KFO	
2988	70AA	CTRT 026				STH U AS,T-2	
298A	27C7	CTRT 027				D1=0	
298C	77A0	CTRT 028				STB D1 AS,T	
298E	25FF	CTRT 029				G1=G1+KFF	
2990	F495	CTRT 030		032	CDE1	BR IF AC=1	
2992	ACA4	CTRT 031		CLOG 052	OP95A	BR	
2994	AC84	CTRT 032	CDE1	CLOG 050	OP95C	BR	
29F8	AB53	CTRT 033	SSOPDE	CICY 045	ADDERR N	BR IF H1=NZ	
29FA	14A3	CTRT 034				G0=G0*-K0A	RST G04 AND G06
29FC	5569	CTRT 035				D0=G1	MOVE LENGTH TO D0
29FE	25F5	CTRT 036				G1=0\$KFO	PUT EBCDIC ZONE IN G1
2A00	C904	CTRT 037		039	EBCDIC	BR IF P1 BIT4=0	TEST FOR NO ASCII
2A02	2555	CTRT 038				G1=0\$K50	PUT IN ASCII ZONE
2A04	5B10	CTRT 039	EBCDIC			RDB T1 U+0	READ FIRST PATTERN
2A06	5BF9	CTRT 040				H1=T1	T1 HAS FILL CHAR
2A08	0F2D	CTRT 041	TSTPAT			Z=H1\$K20	TEST FOR PATTERN CHAR
2A0A	EC96	CTRT 042		048	NOPAT	BR IF HZNZ	BRANCH IF NOT
2A0C	F0B9	CTRT 043		065	DIGSEL	BR IF LZ=0	BR TO DIGIT SELECT
2A0E	CF1B	CTRT 044				Z=H1\$K01	TEST FOR SIG START
2A10	FCB7	CTRT 045		064	SIGST	BR IF LZ=0	
2A12	0F2B	CTRT 046				Z=H1\$K02	TEST FOR FIELD SEP
2A14	F0B1	CTRT 047		055	FLDSEP	BR IF LZ=0	
2A16	E01B	CTRT 048	NOPAT	050	NOFILL	BR IF G0 BIT6=1	TEST S STAT
2A18	7B10	CTRT 049	FILL			STB T1 U+0	STORE FILL
2A1A	5004	CTRT 050	NOFILL			U=U+1	UPDATE ADDRESS
2A1C	26FF	CTRT 051	NEXT			D0=D0+KFF	DECREMENT LENGTH
2A1E	F4A4	CTRT 052		058	FINISH	BR IF AC=0	TEST FOR END
2A20	5F10	CTRT 053				RDB H1 U+0	READ NEXT PATTERN CHR
2A22	AAC8	CTRT 054		041	TSTPAT	BR	
2A30	1427	CTRT 055	FLDSEP			G0=G0*-K22	RST G02 AND G06
2A32	3443	CTRT 056				G0=G0\$K04	SET G05
2A34	AA18	CTRT 057		049	FILL	BR	
2A24	E02D	CTRT 058	FINISH	061	SSCH2	BR IF G0 BIT6=1	TEST S STAT
2A26	E42B	CTRT 059		063	SSCOD2	BR IF G0 BIT2=1	TEST FOR ZERO FIELD
2A28	A8DC	CTRT 060	SSCODX	CICY 003	CHECKX	BR	
2A2C	E428	CTRT 061	SSCH2	060	SSCODX	BR IF G0 BIT2=0	TEST FOR ZERO FIELD
2A2E	AC84	CTRT 062		CLOG 050	OP95C	BR	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2A2A	ACA4	CTR1 063	SSCOD2	CLOG 052	OP95A	BR	
2A36	1443	CTR1 064	SIGST			GO=GO*-K04	RST G05
2A38	C03F	CTR1 065	DIGSEL	068	LODIG	BR IF GO BIT4=1	TEST FOR LU DIGIT
2A3A	A353	CTR1 066		CICY 045	ADDERR N	BR IF D1=NZ	
2A3C	5738	CTR1 067				RDB D1 V+1	READ NEXT SOURCE CHAR
2A3E	4755	CTR1 068	LODIG			G1=D1XL\$G1H	TRNS PACHED TO ZONED
2A40	FCE3	CTR1 069		086	DGTZRO	BR IF LZ=0	SOURCE DIGIT ZERO
2A42	0769	CTR1 070				Z=D1+K60	MASK FOR NON DEC DIGT
2A44	F4E1	CTR1 071		085	NONDEC	BR IF AC=1 **BRANCH	IF GREATER THAN 9
2A46	E05B	CTR1 072		082	ISSSTA	BR IF GO BIT6=1	TEST S STAT
2A48	F05A	CTR1 073		082	ISSSTA	BR IF GO BIT7=0	TEST FOR EDT AND MARK
2A4A	7EE2	CTR1 074				STH H DA,BC	KEEP H
2A4C	5BE9	CTR1 075				H0=T1	KEEP T1
2A4E	2F07	CTR1 076				H1=0	
2A50	2B17	CTR1 077				T1=0\$K11	
2A52	7FA8	CTR1 078				STB H1 AS,T+1	
2A54	70A0	CTR1 079				STH U AS,T+0 **SIGNIF	DIGIT ADDR IN AUXST
2A56	5EB9	CTR1 080				T1=H0	
2A58	5EE2	CTR1 081				RDH H DA,BC	
2A5A	3427	CTR1 082	ISSSTA			GO=GO\$K22	SET G02,G06 FIELD NZ
2A5C	7518	CTR1 083	SSTAT			STB G1 U+1	STORE DIGIT
2A5E	AA6A	CTR1 084		090	NOSGST	BR	
2A60	BC7C	CTR1 085	NONDEC	CCOM 071	FLGDEX	BR	
2A62	E05D	CTR1 086	DGTZRO	083	SSTAT	BR IF GO BIT6=1	TEST S STAT
2A64	7B18	CTR1 087				STB T1 U+1	STORE FILL
2A66	D06B	CTR1 088		090	NOSGST	BR IF GO BIT5=1	TEST FOR SIG START
2A68	34E3	CTR1 089				GO=GO\$K06	SIG START S ON
2A6A	5773	CTR1 090	NOSGST			D1=D1XH	
2A6C	0769	CTR1 091				Z=D1+K60 **MASK FOR NON	DECIMAL DIGIT
2A6E	F4F5	CTR1 092		095	SIGN	BR IF AC=1 ** BRANCH	IF GREATER THAN 9
2A70	148B	CTR1 093				GO=GO\$K08	CHANGE FF STATE
2A72	AA1C	CTR1 094		051	NEXT	BR	
2A74	F77A	CTR1 095	SIGN	098	PLUS	BR IF D1 BIT3=0	
2A76	07FD	CTR1 096				Z=D1\$KFO	
2A78	C4FC	CTR1 097		099	MINUS	BR IF ZNZ	
2A7A	1423	CTR1 098	PLUS			GO=GO*-K02	TURN S OFF
2A7C	27C7	CTR1 099	MINUS			D1=0	
2A7E	3443	CTR1 100				GO=GO\$K04	
2A80	AA1C	CTR1 101		051	NEXT	BR	

 * CROSS REFERENCE FOR CSECT CTRT *

CTR1 003	CICY 237	CICY 238
CTR1 005	CTR1 015	CTR1 020
CTR1 017	CTR1 012	
CTR1 022	CTR1 017	
CTR1 032	CTR1 030	
CTR1 033	CICY 239	CICY 240
CTR1 039	CTR1 037	
CTR1 041	CTR1 054	
CTR1 048	CTR1 042	
CTR1 049	CTR1 057	
CTR1 050	CTR1 048	

* CROSS REFERENCE FOR CSECT CTRT *

CTR1 051	CTR1 094	CTR1 101
CTR1 055	CTR1 047	
CTR1 058	CTR1 052	
CTR1 060	CTR1 061	
CTR1 061	CTR1 058	
CTR1 063	CTR1 059	
CTR1 064	CTR1 045	
CTR1 065	CTR1 043	
CTR1 068	CTR1 065	
CTR1 082	CTR1 072	CTR1 073
CTR1 083	CTR1 086	
CTR1 085	CTR1 071	
CTR1 086	CTR1 069	
CTR1 090	CTR1 084	CTR1 088
CTR1 095	CTR1 092	
CTR1 098	CTR1 095	
CTR1 099	CTR1 097	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCHN 001	T			CHANNEL 0 INTERRUPT ROUTINE	
		DCHN 028	*				THIS IS THE ENTRY POINT FOR
		DCHN 029	*				ALL CHNL 0 I/O INTERRUPTS FROM
		DCHN 030	*				THE BSWI ROUTINE
23F2	56D2	DCHN 031	ENTRY			RDH D DA,BA	RD OUT CHNL 0 INT BUFFER
23F4	1773	DCHN 032				D1=D1*-K07	SET UCW ADR TO HWD 0
23F6	57B5	DCHN 037	NTCNTR			T1=D1XL	T=COOX
23F8	1B83	DCHN 038				T1=T1*-K08 ** AUX B4=1403,B5=2540R,B6=2540P,B7=1052	
23FA	3BB5	DCHN 039				T1=T1\$K80	T=00B4,5,6 OR 7
23FC	5BA0	DCHN 040				RDB T1 AS,T+0 ** T IS POINTED AT --B4,B5,B6,ORB7	
		DCHN 041	*				WHICH IS 1403,2540R,2540P,1052
		DCHN 042	*				UNIT ADDRESSES IN ORDER ABOVE
23FE	5468	DCHN 043				RDH G AS,D+2	READ OUT CHNL STA/OP-FLGS
2400	C251	DCHN 044		082	ZUNSTA	BR IF D04=1	BR IF PCI
2402	B320	DCHN 045		DPTT 089	RESINB	BAL ** GO RST S7,INT	BUFFER BIT AND SEC.
2404	4052	DCHN 046				RDH U DC,9A	READ ZERO'S INTO U
2406	4206	DCHN 047				V=U	MAKE V=0000
2408	D146	DCHN 048		077	INAINT	BR IF G1 BIT5=0	BR IF NOT ACTIVE
240A	C110	DCHN 049		052	NOPCI	BR IF G1 BIT4=0	BR IF NO PCI FLAG
240C	3485	DCHN 050				GO=GO\$K80	SET PCI BIT IN CHANNEL STATUS
240E	1583	DCHN 051				G1=G1*-K08	TURN PCI FLAG OFF
2410	556A	DCHN 052	NOPCI			RDB G1 AS,D-1	RD UNIT STA INTO G1
2412	D119	DCHN 053		056	DEVEND	BR IF G1 BIT5=1	BR IF DEVICE END
2414	C118	DCHN 054		056	DEVEND	BR IF G1 BIT4=0	BR IF NOT CHANNEL END
2416	3C85	DCHN 055				UO=UO\$K80	TURN ON SECONDARY BIT
2418	5666	DCHN 056	DEVEND			D=D-1	SET UCW ADR TO HWD 0
241A	7068	DCHN 057				STH U AS,D+2	STORE 8000 IN UCW HWD 0
241C	7160	DCHN 058				STB U1 AS,D+0	ZERO OUT UNIT STATUS
241E	176B	DCHN 059	PCISS			D1=D1#K06	SET D TO UCW HWD 2
2420	5068	DCHN 060				RDH U AS,D+2	RD UCW HWD 2
2422	5260	DCHN 061				RDH V AS,D+0	RD UCW HWD 3
2424	021E	DCHN 062				RST MMSK K=71	RST LEVEL 1 TRAP LOCKOUT
2426	0C80	DCHN 063	CSWST			RST S4	RST DE. ALONE FLAG
2428	55E9	DCHN 064	DEALON			H0=G1	MOVE UNIT STATUS
242A	5F59	DCHN 065				G1=H1	SAVE CPU ADDR BYTE
242C	54F9	DCHN 066				H1=GO	MOVE CHANNEL STATUS
242E	5B49	DCHN 067				GO=T1	SAVE UNIT ADDRESS
2430	D235	DCHN 068		070	COMTIO	BR IF D05=1	BR IF COMM. TEST I/O
2432	3000	DCHN 069				SET S0	SET RETURN FLAG
2434	2400	DCHN 070	COMTIO			SET MODE K=00	SET CPU MODE
2436	C1C3	DCHN 071		090	STAT	BR IF S4=1	BR IF DE. ALONE
2438	A00E	DCHN 072		CCOM 110	FULCSW	BAL	GO STORE FULL CSW
243A	54B9	DCHN 073	FINISH			T1=GO	RESTORE UNIT ADDRESS
243C	55F9	DCHN 074	FRMFIL			H1=G1	RESTORE CPU ADDRESS BYTE
243E	2135	DCHN 075	USECOM			U1=0\$K30	STRT TO BUILD ADR OF I/O OLD PSW
2440	9C74	DCHN 076		BPSW 005	ENTRYB	BR	BR TO STORE OLD PSW
2446	5560	DCHN 077	INAINT			RDB G1 AS,D+0	RD OUT UNIT STATUS FROM UCW
2448	716A	DCHN 078				STB U1 AS,D-1	CLEAR UNIT STATUS
244A	7060	DCHN 079				STH U AS,D+0	STORE 0000 IN UCW HWD 0
244C	2080	DCHN 080				SET S4	REMEMBER DE. ALONE
244E	A428	DCHN 081		064	DEALON	BR	GO MOVE STATUS FOR STATUS=0,SCSW
2450	5666	DCHN 082	ZUNSTA			D=D-1	DECR D TO UCW BYTE 1
2452	0010	DCHN 083				RST S7	RST CHAN-0 INT

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2454	1583	DCHN 084				G1=G1*-K08	RST PCI FLAG
2456	1413	DCHN 085				GO=GO*-K01	RST INT BUFFER BIT
2458	7468	DCHN 086				STH G AS,D+2	STORE UPDATED CH STA/OP-FLGS
245A	2485	DCHN 087				GO=0\$K80	SET PCI ALONE
245C	2505	DCHN 088				G1=0\$K00	SET 00 UNIT STATUS
245E	A41E	DCHN 089		059	PCISS	BR	
2442	A01E	DCHN 090	STAT	CCOM 123	STATOS	BAL	GO STORE STATUS AND ZERO CSW
2444	A43A	DCHN 091		073	FINISH	BR	GO TAKE INTERRUPT
2162	2400	DCHN 092	MQDRST			SET MODE K=00	
2164	2A07	DCHN 093	CICRTN			TO=0	
2166	C3EA	DCHN 094		096	NOTIPL	BR IF BA BIT4=0	BR IF NOT IPL
2168	875A	DCHN 095		DCLL 021	IPLSTP	BR	GO TO IPL STOP
216A	A8DE	DCHN 096	NOTIPL	CICY 004	CHECK	BR	
30FA	2400	DCHN 097	CC3SET			SET MODE K=00	
30FC	2C75	DCHN 098	CC3			PO=0\$K70	
30FE	A164	DCHN 099		093	CICRTN	BR	

 * CROSS REFERENCE FOR CSECT DCHN *

DCHN 031	BSWI 102		
DCHN 052	DCHN 049		
DCHN 056	DCHN 053	DCHN 054	
DCHN 059	DCHN 089		
DCHN 064	DCHN 081		
DCHN 070	DCHN 068		
DCHN 073	DCHN 091		
DCHN 074	FINT 046		
DCHN 075	DCLT 045		
DCHN 077	DCHN 048		
DCHN 082	DCHN 044		
DCHN 090	DCHN 071		
DCHN 092	DCLC 191		
DCHN 093	DCHN 099		
DCHN 096	DCHN 094		
DCHN 097	DCLA 116	DCLA 134	
DCHN 098	DCLA 047		

DCLA -- I/O INSTRUCTIONS ROUTINE

ENTRY POINTS

- IOINST -- MAJOR ENTRY POINT FOR DCLA. ENTRY IS FROM I-CYCLES AFTER ONE OF THE FOLLOWING I/O INSTRUCTIONS IS DECODED -- START I/O, TEST I/O, HALT I/O, OR TEST CHANNEL.
- UADRID -- FROM IPL ROUTINE. IPL IS A SIMULATED START I/O, AND USES -DCLA- FOR THE SAME PURPOSE, I.E., TO GET THE UNIT ADDRESS.
- DYPADD, HOHU, STORO -- FROM PR-KB OR ICA ROUTINES. THESE ENTRIES ARE USED TO SHARE THE TESTING OF DEVICE FLAGS WHEN A PR-KB OR COMMUNICATIONS HALT I/O INSTRUCTION IS IN PROCESS.

DESCRIPTION

I/O INSTRUCTIONS ARE DECODED IN THE I-CYCLES ROUTINE, -CICY-, CAUSING A BRANCH TO -DCLA- .
 IN -DCLA-, A CHECK IS MADE TO ENSURE THAT PRIVILEGED OPS ARE PERMISSABLE, I.E., SUPERVISOR STATE. THE CONDITION CODE IS INITIALIZED TO 0 -- THIS IS CHANGED LATER IF THE I/O UNIT IS BUSY. FOR START I/O ONLY, THE CAW IS FETCHED AND CHECKED. FOR ALL I/O INSTRUCTIONS, THE CHANNEL IS IDENTIFIED. WHEN CHANNEL 1 IS IDENTIFIED, THE ROUTINE BREAKS OUT TO EITHER THE FILE OR BURST CHANNEL ROUTINES IF THESE FEATURES ARE PRESENT.
 WHEN CHANNEL 0 IS IDENTIFIED, THE ACTIVE BIT IS SET ON IN THE REGISTER USED TO HOLD THE NEW FLAGS/OP BYTE. THE UCW ADDRESS IS GENERATED, THEN THE MICROPROGRAM BRANCHES TO ANOTHER ROUTINE ON THE BASIS OF THE OP CODE, DEVICE TYPE, AND ACTIVITY STATUS.

OBJECTIVES

1. FOR START I/O, FETCH AND CHECK CAW.
2. GENERATE UCW ADDRESS.
3. EXIT EITHER TO DO THE OPERATION REQUIRED, OR TO RETURN TO I-CYCLES AFTER SETTING THE CONDITION CODE.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLA 001	T			I/O INSTRUCTIONS	H.E.BERKEBILE
0CB8	10CE	DCLA 002	IOINST			RST S K=FC	RST S REG BITS 0 THRU 5
0CBA	F93E	DCLA 003		005	SUPVSR	BR IF P17=0	BR IF SUPERVISOR MODE
0CBC	B35A	DCLA 004		CICY 068	PRIVOP	BR	
0CBE	2C07	DCLA 005	SUPVSR			PO=0	SET COND. CODE 0
0CC0	2B45	DCLA 006				T1=0\$K40	
0CC2	3B83	DCLA 007				T1=T1\$K08	SET UP CAW ADDRESS
0CC4	46A6	DCLA 008				D=T	D=C048 ---START OF UCW ADDR
0CC6	17FD	DCLA 009				D1=D1\$KFO	D=00B8-----
0CC8	801F	DCLA 010		011	IOOP N	N=GO BITS67	BR ON OP
0F80	8798	DCLA 011	IOOP 0	063	STRTIO	BR	START I/O
0F82	87AE	DCLA 012	IOOP 1	089	UADRID	BR	TEST I/O
0F84	87AE	DCLA 013	IOOP 2	089	UADRID	BR	HALT I/O
0F86	5009	DCLA 014	IOOP 3			U0=U0	TEST CHANNEL
0F88	C490	DCLA 018		044	NOTCHO	BR IF ZNZ	BR IF NOT CHNL 0
0F8A	C78F	DCLA 020	CHNL0T	042	CC1T	BR IF BA BIT0=1	BR IF CHNL 0 IB IS ON
0F8C	B362	DCLA 041	CC0T	CCOM 154	CC0B	BR	SET CC=0
0F8E	ADF4	DCLA 042	CC1T	CCOM 157	CC1B	BR	SET CC=1
0F90	001B	DCLA 044	NOTCHO			Z=U0\$K01	
0F92	C497	DCLA 045		049	CHNL1T	BR IF Z=0	BR IF NOT CHNL 1
0F94	B0FC	DCLA 047	CC3T	DCHN 098	CC3	BR	BR TO SET CDN CODE 3
0F96	2486	DCLA 049	CHNL1T			SET MODE K=38	SET CHANNEL MODE

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0F98	D88F	DCLA 050		042	CCIT	BR IF GS BIT5=1	BR IF CHAN INT. IS ON
0F9A	2480	DCLA 052				SET MODE K=08	SET FILE MODE
0F9C	E98F	DCLA 053		042	CCIT	BR IF FF16=1	BR IF ANY FILE INT. IS ON
0F9E	5632	DCLA 058				RDH D DA,8E	GET STATS
0FA0	C60C	DCLA 059		041	CCOT	BR IF DO BIT0=0	BR IF ACTIVE OFF
0FA2	AC74	DCLA 060		CCOM 161	CC2B	BR	
		DCLA 062	*			BEGIN READOUT AND CHECK OF CAW FOR START I-O.	
0798	5EB8	DCLA 063	STRTIO			RDH H T+2	READ 1ST HW OF CAW
079A	7EF2	DCLA 065				STH H DA,8E	SAVE THE KEY
079C	0EF5	DCLA 066				Z=HO*-KFO	MASK LOW 4 BITS OF KEY BYTE
079E	FOA3	DCLA 067		069	HIADDR	BR IF LZ=0	BR IF LD 4 BITS OF KEY BYTE=0
07A0	B0C4	DCLA 068	PROG 0	DCLR 008	PGESTT	BR	BAD CAW--PROG CHECK
07A2	AB3F	DCLA 069	HIADDR	068	PRCG N	BR IF H1=NZ	BR IF HIGH ADDR NON ZERO
07A4	52BA	DCLA 075	RDCAW			RDH V T-2	READ CCW ADDRESS
07A6	0249	DCLA 083				Z=VO+K40	
07A8	F4A1	DCLA 085		068	PRCG 0	BR IF AC=1	BR IF CCW ADR OVER MEMORY SIZE
07AA	0383	DCLA 086				Z=V1*-K08	CHK FOR 8 BOUNDARY
07AC	FOA0	DCLA 087		068	PRCG 0	BR IF LZNZ	
		DCLA 088	*			IDENTIFY THE CHANNEL FOR START I-O, TEST I-O, AND HALT I-O.	
07AE	5009	DCLA 089	UADRIO			UO=UO	
07B0	C4B4	DCLA 090		093	NCHNLO	BR IF ZNZ	BR IF NOT CHNL 0
07B2	8888	DCLA 091		124	CHNLO	BR	
07B4	001B	DCLA 093	NCHNLO			Z=UO#K01	
07B6	C4C6	DCLA 094		116	CC3S	BR IF ZNZ	BR IF NOT CHNL 1
07B8	5632	DCLA 099	CHLFIL			RDH D DA,8E	RD STATS INTO D
07BA	2BA1	DCLA 101				T1=0-K0A	T=00F5
07BC	5FAA	DCLA 102				RDB H1 AS,T-1	RD OUT CONTROL UNIT ADR OF FILE
07BE	6F11	DCLA 103				H1=H1#U1	
07C0	E0C5	DCLA 104		113	FILE	BR IF HZ=0	BR IF FILE
07C2	8003	DCLA 108		109	BRSTOP N	N=GO BITS67	BURST CHNL--BR ON DP BYTE
0100	846C	DCLA 109	BRSTOP 0	DCLB 004	CCWBUR	BR	START I/O-BR TO CCW FETCH RTNE
0102	AF02	DCLA 110	BRSTOP 1	DCLT 003	TESTIO	BR	TEST I/O
0104	A340	DCLA 111	BRSTOP 2	DCLH 004	ENTRY	BR	HALT I/O
07C4	847A	DCLA 113	FILE	FILE 008	ENTRY	BR	BR TO 2311 ROUTINE
07C6	BOFA	DCLA 116	CC3S	DCHN 097	CC3SET	BR	
		DCLA 117	*			BEGIN CONSTRUCTION OF UCW ADDRESS IN DO/D1. FOR NATIVE PRINTER, 00C8.	
		DCLA 118	*			FOR NATIVE READER, 00D8. FOR NATIVE PUNCH, 00E8. FOR NATIVE TYPE-	
		DCLA 119	*			WRITER, 00F8. FOR NATIVE COMMUNICATIONS, 80XX. FOR NON-NATIVE CHAN-	
		DCLA 120	*			NEL, 20XX. XX IS DERIVED FROM UNIT ADDRESS, THUSLY--FOR NATIVE COMM-	
		DCLA 121	*			UNICATIONS, XX=U3,U4,U5,U6,U7,0,0,0. FOR OTHER (NON-NATIVE) SINGLE	
		DCLA 122	*			SUBCHANNELS, XX=U4,U5,U6,U7,U3,0,0,0. FOR SHARED SUBCHANNELS, XX=	
		DCLA 123	*			0,U1,U2,U3,0,0,0,0.	
0888	2543	DCLA 124	CHNLO			G1=0#K04	SET ACTIVE BIT
088A	4E66	DCLA 125				H=D	H=00B8
088C	1FCB	DCLA 126				H1=H1#K0C	H=00B4
088E	2710	DCLA 127	RDNADR			D1=D1+K10	D=C0C8----1ST UCW ADDRESS
0890	5BE8	DCLA 128				RDB T1 AS,H+1 ** AUX-B4=PRINT,READER,PUNCH,1052	
0892	6B11	DCLA 129				T1=T1#U1	CHECK FOR
0894	C49B	DCLA 130		201	TESACT	BR IF Z=0	MATCH
0896	DB0F	DCLA 131		127	RDNADR	BR IF H15=1	BR IF NOT FINISHED
		DCLA 132	*			AT THIS POINT, EITHER COMMUNICATIONS OR NON-NATIVE.	
0898	BOFA	DCLA 134	NOTIPL	DCHN 097	CC3SET	BR	
08DE	889A	DCLA 183	OPERNL	201	TESACT	BR	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLA 199	*			FORMATION OF THE UCW ADDRESS IS COMPLETE. NEXT, BREAK OUT TO OTHER	
		DCLA 200	*			ROUTINES ON THE BASIS OF OP CODE, DEVICE TYPE, ACTIVITY STATUS, ETC.	
089A	5E60	DCLA 201	TESACT			RDH H AS,D+0	RD UCW FIRST HW
089C	E032	DCLA 202		242	NOHALB	BR IF G06=0	BR IF NOT HALT I/O
089E	AC78	DCLA 215	HALTRP	ERRQ 083	RDCHST	BAL ** GO GET CHAN. AND UNIT STATUS IN G,H	
08A0	07FD	DCLA 216				Z=D1DKFO	LOOK FOR 1052
08A2	E0CE	DCLA 217		219	DYPADD	BR IF HZNZ	BR IF OTHER NATIVE
08A4	8EAC	DCLA 218		DYPE 210	HLTTW	BR	
08CE	D526	DCLA 219	DYPADD	223	VANSOI	BR IF G11=0	BRANCH IF NOT CHAINED
08D0	15C5	DCLA 220	HOHO			G1=G1*-KCO	RST CDA AND CC FLAGS
08D2	7460	DCLA 221				STH G AS,D+0	STORE UPDATED CH STAT IN UCW
08D4	88AC	DCLA 222		226	STORO	BR	
08A6	C42C	DCLA 223	VANSOI	226	STORO	BR IF G00=0	BR IF NO SECONDARY
08A8	D131	DCLA 224		228	GOBACK	BR IF G15=1	BR IF ACTIVE
08AA	DA31	DCLA 225		228	GOBACK	BR IF H05=1	BR IF DEVICE END
08AC	4E52	DCLA 226	STORO			RDH H DC,9A	READ ZERO CONSTANT INTO H
08AE	A030	DCLA 227		CCOM 135	STATUS	BR	STURE STATUS CSW
08B0	B362	DCLA 228	GOBACK	CCOM 154	CCOB	BR	SET CCO
08B2	F045	DCLA 242	NOHALB	277	TIOBR	BR IF G07=1	BR IF TEST I/O
08B4	DB43	DCLA 243		285	CC2	BR IF H1 BIT5=1	BR ON ACTIVE BIT
08B6	CE4D	DCLA 252		284	NOTTES	BR IF H00=1	BR IF SEC. ON
0888	5669	DCLA 256	CCHCCW			DO=DO	
08BA	E0BF	DCLA 257		259	DCLBBR	BR IF HZ=0	BR IF NATIVE
08BC	81F8	DCLA 258		310	BURKEY	BAL	GO SET UP THE KEY
08BE	3773	DCLA 259	DCLBBR			D1=D1\$K07	SET D TO UCW HWD 0
08C0	93F4	DCLA 261		DCLB 077	RDHW1	BR	
08C4	DB4A	DCLA 277	TIOBR	283	SETIT	BR IF H15=0 ** BR IF NAT.IS NOT ACTIVE	
08C6	CE42	DCLA 281		285	CC2	BR IF H00=0	SET BUSY IF NO CHAN.END
08C8	DF43	DCLA 282		285	CC2	BR IF H11=1	BR IF STILL CHAINED--SET BUSY
08CA	3613	DCLA 283	SETIT			DO=D0\$K01	REMEMBER TEST IO FOR NATIVES
08CC	8711	DCLA 284	NOTTES	303	INASIO N	N=D1 BITS23 ** GO TO NAT.TEST I/O---START-SEC. ON	
08C2	AC74	DCLA 285	CC2	CCOM 161	CC2B	BR	SET CC=2
		DCLA 301	*			BREAKOUT BY DEVICE, FOR START I-O, ACTIVE=0, SECONDARY=1.	
0880	8948	DCLA 303	INASIO 0	DPTT 006	TEST00	BR	PR,SEC BIT=1 START I/O
0882	9EAA	DCLA 305	INASIO 1	ERRQ 004	REDUC	BR	RD,SEC BIT=1,START IO OR TIO
0884	9EAA	DCLA 306	INASIO 2	ERRQ 004	REDUC	BR	PU,SEC BIT=1,START IO OR TIO
0886	96F4	DCLA 308	INASIO 3	DYPE 191	TWSEC7	BR	TW,SEC BIT=1,START IO
01F8	52FF	DCLA 310	BURKEY			H1=STPO	H1=CURRENT CPU KEY
01FA	526C	DCLA 311				ISK STPO AS,D+2	READ OUT CHNL/COMM KEYS
01FC	52EF	DCLA 312				HO=STPO	HO= CHNL/COMM KEYS
01FE	51F2	DCLA 313				RDB U1 DA,BE	READ OUT NEW KEY
0200	C6C6	DCLA 317		325	MPXKEY	BR IF D00=0	BR IF MPX CHNL
0202	41E5	DCLA 319				HO=U1XL\$HOH	OR OLD MPX AND NEW COMM KEYS
0204	C609	DCLA 323		326	KEYOR	BR IF D00=1	UNC BR
0206	41EB	DCLA 325	MPXKEY			HO=U1H+HOL	OR OLD COMM AND NEW MPX KEYS
0208	42EF	DCLA 326	KEYOR			STPO=HO	MOVE UPDATED KEY TO STPO
020A	536C	DCLA 327				SSK STP1 AS,D+1	STORE UPDATED KEY IN STP1
020C	42FF	DCLA 328				STPO=H1	RESTORE CPU KEY
020E	128E	DCLA 329				RTN	

 * CROSS REFERENCE FOR CSECT DCLA *

DCLA 002	CICY 221	CICY 222	CICY 223	CICY 224
DCLA 005	DCLA 003			
DCLA 011	DCLA 010			
DCLA 041	DCLA 059			
DCLA 042	DCLA 020	DCLA 050	DCLA 053	
DCLA 044	DCLA 018			
DCLA 049	DCLA 045			
DCLA 063	DCLA 011			
DCLA 068	DCLA 069	DCLA 085	DCLA 087	
DCLA 069	DCLA 067			
DCLA 089	DCLA 012	DCLA 013	DCLL 018	
DCLA 093	DCLA 090			
DCLA 109	DCLA 108			
DCLA 113	DCLA 104			
DCLA 116	DCLA 094			
DCLA 124	DCLA 091			
DCLA 127	DCLA 131			
DCLA 201	DCLA 130	DCLA 183		
DCLA 219	DCLA 217	DYPE 225		
DCLA 220	DYPE 220			
DCLA 223	DCLA 219			
DCLA 226	DCLA 222	DCLA 223	DYPE 216	
DCLA 228	DCLA 224	DCLA 225		
DCLA 242	DCLA 202			
DCLA 259	DCLA 257			
DCLA 277	DCLA 242			
DCLA 283	DCLA 277			
DCLA 284	DCLA 252			
DCLA 285	DCLA 243	DCLA 281	DCLA 282	
DCLA 303	DCLA 264			
DCLA 310	DCLA 258	DCLB 011		
DCLA 325	DCLA 317			
DCLA 326	DCLA 323			

CCW FETCH AND CHECK ROUTINE

THE -DCLB- ROUTINE READS AND CHECKS THE CCW'S FOR IPL, START I/O, COMMAND CHAINING, AND DATA CHAINING. THE ROUTINE IS VERSIONED ACCORDING TO THE CHANNEL FEATURE SELECTED FOR THE CONTROL-STORAGE LOAD, I.E., EITHER CHANNEL 0 -- BYTE MODE, OR CHANNEL 1 -- BURST MODE.

ENTRIES

CCWBUR -- BURST MODE ENTRY FROM START I/O OR IPL. THIS PORTION OF THE ROUTINE SETS THE INITIAL SELECT LATCH AND SETS AN INDICATOR BIT IF THE DEVICE IS BUFFERED.

COMCHN -- COMMUNICATIONS CHAINING ENTRY.

CHAIN -- NATIVE I/O OR BYTE MODE CHAINING ENTRY.

CDACUM -- BYTE MODE DATA CHAINING ENTRY.

ENTRY POINTS CONTINUED

SKIP -- BURST MODE ENTRY TO INCREMENT CCW ADDRESS BY 8 BEFORE CHAINING.

RDHWIX -- BURST MODE COMMAND CHAINING ENTRY.

RDHW1 -- BYTE MODE ENTRY FROM START I/O OR IPL. BURST MODE ENTRY FOR DATA CHAINING WHEN NEXT CCW IS A TIC COMMAND.

NOTIC -- BURST MODE ENTRY FROM LOW-PRIORITY TRAP ROUTINE AFTER DETECTION OF AN INCORRECT TIC ADDRESS. THE CCW ADDRESS IS INCREMENTED IN -DCLB- BEFORE BRANCHING TO THE -DCLR- ERROR ROUTINE.

BSTCDA -- ENTRY FROM -DCLR- ERROR ROUTINE TO PERMIT RETURN FROM TRAP AFTER DATA CHAINING ERROR.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLB 001	T			CCW FETCH AND CHECK ROUTINE	H.E.BERKEBILE
		DCLB 003	*			BURST MODE ENTRY FROM START I/O	
046C	C642	DCLB 004	CCWBUR	006	INACT	BR IF DO BIT0=0	BR IF NOT ACTIVE
046E	AC74	DCLB 005	ACTBON	CCOM 161	CC2B	BR	SET CC=2
0442	2045	DCLB 006	INACT			UO=0\$K40	
0444	7022	DCLB 007				STH U DA,8C	STORE UNIT IDENT HWD IN AUX008C
0446	2625	DCLB 009				DO=0\$K20	SET DO TO BUMP 2
0448	2787	DCLB 010				D1=0\$K88	
044A	81F8	DCLB 011		DCLA 310	BURKEY	BAL	BAL TO STORE KEY
044C	2486	DCLB 013				SET MODE K=38	
044E	2B80	DCLB 014				SET GA K=08	SET INL SELECT LTCH
0450	7212	DCLB 015				STH V DA,8A	STORE NEXT CCW ADR IN AUX008A
0452	24F6	DCLB 016				SET MODE K=3F	SET CHNL MODE AND ZONE
0454	5622	DCLB 017				RDH D DA,8C	RD UNIT IDENT HWD INTO D
0456	2407	DCLB 018				GO=0	SET UP BUFFERED
0458	2585	DCLB 019				G1=0\$K80	DEVICE ADDRESSES TABLE
045A	3543	DCLB 020				G1=G1\$K04	ADDRESS - 0084
045C	5148	DCLB 021	NOTFIN			RDB U1 AS,G+1	READ OUT BUFFERED DEVICE ADDRESS
045E	6171	DCLB 022				U1=U1#D1	LOOK FOR EQUAL ADDRESS
0460	C4F1	DCLB 023		029	MATCH	BR IF Z=0	BR IF EQUAL ADDR
0462	C15C	DCLB 024		021	NOTFIN	BR IF G1 BIT4=0	BR IF ADR NOT REACHED 0088 OR 98
0464	25CB	DCLB 025				G1=G1+K0C	CHANGE ADDR TO 0094 OR 00A4
0466	F55D	DCLB 026		021	NOTFIN	BR IF G1 BIT3=1	BR IF ADDR IS 0094
0468	25C7	DCLB 027				G1=0	HAVE REACHED END AND NO MATCH
046A	8472	DCLB 028		030	BSTCC	BR	CONTINUE WITH CCW FETCH
0470	5651	DCLB 029	MATCH			G1=DOX	G1=04 ACTIVE

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0472	5212	DCLB 030	BSTCC			RDH V DA,8A	READ OUT NEXT CCW ADR
0474	93F4	DCLB 031		077	RDHW1	BR	
		DCLB 033	*			NATIVE IO AND BYTE MODE CHANNEL CHAIN ENTRY	
		DCLB 034	*			ENTRY TO FETCH A NEW CCW UPON DATA OR COMMAND CHAINING. DO/D1 CONTAINS UCW ADDRESS, G1 CONTAINS FLAG AND OP BITS OF PREVIOUS UCW.	
		DCLB 035	*				
13CA	1683	DCLB 039	CHAIN			DO=DO*-K08	
13CC	3613	DCLB 040	CCWCDA			DO=DO\$K01	SET CAME FRM CHAINING FLAG
13CE	3773	DCLB 041	CDACOM			D1=D1\$K07	SET D1 TO UCW BYTE 7
13D0	5260	DCLB 042				RDH V AS,D+0	READ NEXT CCW ADR
13D2	D25A	DCLB 043		054	RDHWIX	BR IF D05=0	BR IF NO COMMU. STAT MOD
13D4	1643	DCLB 044				DO=DO*-K04	TURN OFF INCR CCW+8 BIT
13D6	3363	DCLB 045	SKIP			V1=V1\$K06	INCREMENT CCW
13D8	6224	DCLB 046				V=V+2	ADR BY 8
13DA	0249	DCLB 054	RDHWIX			Z=V0*K40	
13DC	F4C3	DCLB 056		106	PRGERC	BR IF AC=1	BR IF CCW ADR OVER MEMORY
13DE	5038	DCLB 057				RDH U V+2	READ FIRST CCW HW
13E0	0C8B	DCLB 058				Z=U0\$K08	
13E2	F0FA	DCLB 059		080	NOTIC	BR IF LZNZ	BR IF NO TIC
		DCLB 060	*			COMMAND CODE IS TIC. CHECK AND EXECUTE.	
13E4	5119	DCLB 061				U1=U1	
13E6	C4C4	DCLB 062		107	PRGERB	BR IF ZNZ	BR IF CCW BYTE 1=NZ
13E8	5030	DCLB 063				RDH U V+0	READ CCW SECOND HW
13EA	0049	DCLB 071				Z=U0+K40	
13EC	F4C5	DCLB 073		107	PRGERB	BR IF AC=1	BR IF DATA ADR OVER MEMORY
13EE	0183	DCLB 074				Z=U1*-K08	
13F0	F0C4	DCLB 075		107	PRGERB	BR IF LZNZ	CHK FOR VALID TIC ADR
13F2	4206	DCLB 076				V=U	
13F4	5038	DCLB 077	RDHW1			RDH U V+2	READ FIRST CCW HW
13F6	0C8B	DCLB 078				Z=U0\$K08	
13F8	F0C5	DCLB 079		107	PRGERB	BR IF LZ=0	CHK FOR VALID TI
13FA	5119	DCLB 080	NOTIC			U1=U1	
13FC	C4C4	DCLB 081		107	PRGERB	BR IF ZNZ	BR IF TIC CCW BYTE 1=NZ
13FE	5049	DCLB 082				GO=U0	MOVE COMMAND TO GO
1400	6224	DCLB 083				V=V+2	INCREMENT CCW ADR
1402	5038	DCLB 084				RDH U V+2	READ CCW THIRD HW
1404	C083	DCLB 085				Z=U0*-K08	CHK FOR VALID FLAG
1406	F094	DCLB 086		093	PRGERA	BR IF LZNZ	CHK FOR VALID FLAGS
1408	C51F	DCLB 087		114	CDABR	BR IF G1 BIT0=1	CHK FOR DATA CHAIN
140A	1537	DCLB 088				G1=G1*-K33	G1=0000X100
140C	F031	DCLB 089	ABCD	110	OUTCMD	BR IF GO BIT7=1	DECODING OF
140E	E033	DCLB 090		111	INFCMD	BR IF GO BIT6=1	THE CCW
1410	C035	DCLB 091		112	INBCMD	BR IF GO BIT4=1	COMMAND
1412	D033	DCLB 092		111	INFCMD	BR IF GO BIT5=1	BYTE
1414	6224	DCLB 093	PRGERA			V=V+2	INCR V TO CORRECT NXT CCW ADR
1416	D639	DCLB 095	PRGERR	103	BSTPER	BR IF D0 BIT1=1	BR IF BURST CHNL
1418	3763	DCLB 096				D1=D1\$K06	SET D TO UCW HWD 3
141A	7260	DCLB 100				STH V AS,D+0	STORE NEXT CCW ADR
141C	8C82	DCLB 101	PRGERX	DCLR 002	PRGERR	BR	
1438	7212	DCLB 103	BSTPER			STH V DA,8A	STORE UPDATED NXT CCW ADR
143A	8C82	DCLB 104		DCLR 002	PRGERR	BR	
13C2	6224	DCLB 106	PRGERC			V=V+2	UPDATE V
13C4	6224	DCLB 107	PRGERB			V=V+2	TO CORRECT
13C6	6224	DCLB 108				V=V+2	NEXT CCW ADR

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
13C8	9414	DCLB 109		093	PRGERA	BR	
1430	153B	DCLB 110	OUTCMD			G1=G1#K03	SET OP BITS TO OUTPUT
1432	151B	DCLB 111	INFCMD			G1=G1#K01	SET OP BITS TO INP FW
1434	153B	DCLB 112	INBCMD			G1=G1#K03	SET OP BITS TO INPBKW
1436	9420	DCLB 113		115	FLAGOR	BR	
141E	2405	DCLB 114	CDABR			GO=0\$K00	SET CD INDICATION
1420	15F5	DCLB 115	FLAGOR			G1=G1*-KFO	CLEAR HIGH FLAGS
1422	6505	DCLB 116				G1=G1\$UO	OR THE OLD AND NEW FLAGS
1424	5038	DCLB 117	CHKFLG			RDH U V+2	READ CCW FOURTH HW
1426	40C6	DCLB 118				U=U	
1428	C497	DCLB 119		095	PRGERR	BR IF Z=0	BR IF CCW COUNT=0000
142A	D63D	DCLB 121		126	STBURS	BR IF DO BIT1=1	BR IF BURSTCH
142C	726A	DCLB 123				STH V AS,D-2	STORE NEXT CCW ADR
142E	943E	DCLB 124		128	DECV	BR	XXXXXXXXXXMAKE ADDRESSABLE
143C	7212	DCLB 126	STBURS			STH V DA,8A	STORE NEXT CCW ADR
143E	6226	DCLB 128	DECV			V=V-2	DECREMENT V TO HWD 1
1440	1343	DCLB 129				V1=V1*-K04	OF CURRENT CCW
		DCLB 130	*			LAST TO BE FETCHED IS THE DATA ADDRESS FIELD OF THE NEW CCW.	
1442	5230	DCLB 131				RDH V V+0	READ CCW SECOND HW
		DCLB 132	*			CCW IS NOW ALL FETCHED, READY TO BE TURNED OVER TO USING I-O ROUTINE.	
		DCLB 133	*			CCW DATA DISTRIBUTED AS FOLLOWS--	
		DCLB 134	*			GO CONTAINS COMMAND CODE (=0 IF DATA CHAINED TO PRESENT CCW),	
		DCLB 135	*			G1 CONTAINS FLAG/OP BYTE FOR NEW UCW,	
		DCLB 136	*			V0/V1 CONTAINS INITIAL DATA ADDRESS,	
		DCLB 137	*			U0/U1 CONTAINS INITIAL COUNT,	
		DCLB 138	*			D0/D1 CONTAINS UCW ADDRESS+2,	
		DCLB 139	*			DO BIT 7=1 IF CURRENT CCW IS NOT FIRST IN CURRENT CHAIN, AND	
		DCLB 140	*			NEXT CCW ADDRESS IS IN AUX STORAGE.	
1444	D659	DCLB 142		190	CHCCW	BR IF D01=1	BR IF BURST
1446	C54A	DCLB 144		165	NOTNAT	BR IF G1 BIT0=0	BR IF NO CDA FLAG
1448	1565	DCLB 145				G1=G1*-K60	ZERO OUT CC AND SLI FLAGS
144A	1763	DCLB 165	NOTNAT			D1=D1*-K06	
144C	756A	DCLB 166				STB G1 AS,D-1	STORE UCW FLAGS,OP
144E	6666	DCLB 174				D=D-2	
1450	F25F	DCLB 175		179	CHAKEY	BR IF D07=1	BR IF FROM CHAIN
1452	5FF2	DCLB 176				RDB H1 DA,BE	GET SAVED CAW KEY-START I/O
1454	7F60	DCLB 177				STB H1 AS,D	STORE KEY IN UCW OOX6
1456	9460	DCLB 178		180	SETKEY	BR	
145E	5F60	DCLB 179	CHAKEY			RDB H1 AS,D	GET ASSIGNED KEY FROM UCW
1460	6664	DCLB 180	SETKEY			D=D+2	
1462	42FF	DCLB 181				STPO=H1	SET KEY FOR NATIVE
1464	8721	DCLB 183		209	NATSIO N	N=D1 BITS23	BR TO DIF NAT IO STIO
		DCLB 184	*			IF CHANNEL START I-O, GO ON TO INITIAL SELECTION.	
		DCLB 185	*			IF CHANNEL CMD CHAIN, GO ON TO INITIAL SELECTION, THEN RTN FROM TRAP.	
		DCLB 186	*			IF CHANNEL DATA CHAIN, OR NATIVE CMD OR DATA CHAIN, RETURN FROM TRAP.	
1458	A041	DCLB 190	CHCCW	199	NOTCDA N	BR IF GO=NZ	BR IF CMD IS NZ (IF NOT CDA)
145A	1D00	DCLB 191	BSTCDA			RST GB K=80	RST CDA REQ LTCH
145C	0244	DCLB 195				RTN G MMSK2=0	RETURN FROM TRAP
1040	8AEE	DCLB 199	NOTCDA 0	DCLC 014	ENTRY	BR	EXIT TO INITIAL SELECTION
		DCLB 207	*			IF NATIVE START I-O, BRANCH TO INDIVIDUAL ROUTINES.	
1020	9CFC	DCLB 209	NATSIO 0	DPTS 042	PRCIO	BR	PRINTER START I/O SEC BIT=0
1022	5812	DCLB 211	NATSIO 1	ERDR 053	NRDR	BR	RDR START IO,SEC BIT=0
1024	5812	DCLB 212	NATSIO 2	ERDR 053	NRDR	BR	PCH START IO,SEC BIT=0

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1026	A4A2	DCLB 214	NATSIO 3	DYPE 113	TWOFF7	BR	TW,START IO,SEC BIT=0
		DCLB 215	*	*****		END OF DCHB (CCW FETCH ROUTINE).	*****

						* CROSS REFERENCE FOR CSECT DCLB *	

DCLB 004		DCLA 109					
DCLB 006		DCLB 0C4					
DCLB 021		DCLB 024	DCLB 026				
DCLB 029		DCLB 023					
DCLB 030		DCLB 028					
DCLB 039		DPTQ 037	DYPE 155	ECOL 070			
DCLB 045		DCLC 148					
DCLB 054		DCLB 043	DCLC 149				
DCLB 077		DCLA 261	DCLB 031	DCLC 023			
DCLB 080		DCLB 059	DCLC 026				
DCLB 093		DCLB 086	DCLB 109				
DCLB 095		DCLB 119					
DCLB 103		DCLB 055					
DCLB 106		DCLB 056	DCLC 025				
DCLB 107		DCLB 062	DCLB 073	DCLB 075	DCLB 079	DCLB 081	
DCLB 110		DCLB 089					
DCLB 111		DCLB 090	DCLB 092				
DCLB 112		DCLB 091					
DCLB 114		DCLB 087					
DCLB 115		DCLB 113					
DCLB 126		DCLB 121					
DCLB 128		DCLB 124					
DCLB 165		DCLB 144					
DCLB 179		DCLB 175					
DCLB 180		DCLB 178					
DCLB 190		DCLB 142					
DCLB 191		DCLR 047					
DCLB 199		DCLB 190					
DCLB 209		DCLB 183					

DCLC DESCRIPTIVE TEXT

CHANNEL INITIAL SELECTION

THIS ROUTINE PROVIDES THE INITIAL SELECTION SEQUENCE FOR DEVICES ATTACHED TO THE STANDARD INTERFACE. THE ROUTINE IS VERSIONED ACCORDING TO THE CHANNEL FEATURE, EITHER BYTE CHANNEL-0, OR BURST CHANNEL-1.

BURST CHANNEL ENTRY POINTS

- ENTRY -- INITIAL SELECTION ENTRY, OR RESELECTION ENTRY IF CHAINING, FOR START I/O OR IPL AFTER COMPLETING CCW FETCH IN -DCLB- ROUTINE.
- NUPCIB -- ENTRY FOR I/O INTERRUPT OR INACTIVE TEST I/O TO GET DEVICE ON THE INTERFACE.
- HLIENT -- INACTIVE HALT I/O ENTRY TO GET DEVICE ON THE INTERFACE.
- ADRABC -- FROM CHANNEL LOW-PRIORITY-TRAP STATUS ROUTINE -DCLE- TO CONTINUE TESTING STATUS. FROM THIS POINT, THE ROUTINE CAN EITHER COMMAND CHAIN, TRY TO LOAD THE INTERRUPT BUFFER, OR STORE THE CSW.
- ADR11A -- FROM -DCLE- STATUS ROUTINE TO TRY LOADING THE INTERRUPT BUFFER OR TO STORE THE CSW.
- TRYIBS -- FROM -DCLE- STATUS ROUTINE TO TRY LOADING THE INTERRUPT BUFFER.
- ADR22A -- ENTRY TO DO A TIMEOUT ON FALL OF OPERATIONAL-IN, THEN EITHER RETURN FROM TRAP OR GO TO I-CYCLES.

BYTE CHANNEL ENTRY POINTS

- ENTRY -- INITIAL SELECTION ENTRY, OR RESELECTION ENTRY IF CHAINING, FOR START I/O OR IPL AFTER COMPLETING CCW FETCH IN -DCLB- ROUTINE.
- BEGIN -- ENTRY TO SELECT THE DEVICE FOR HALT I/O, I/O INTERRUPT, OR WHEN A CHANNEL ERROR IS DETECTED.
- NOCHE -- ENTRY FROM LOW-PRIORITY-TRAP STATUS ROUTINE -DCLE- TO EITHER COMMAND CHAIN OR SET UP FOR INTERRUPT.
- IPLCHK STATUS -- ENTRY FROM -DCLE- STATUS ROUTINE TO SET UP FOR INTERRUPT.
- CSWFRM -- FROM INTERRUPT ROUTINE OR TEST I/O ROUTINE TO PUT TOGETHER CSW.
- ACTPCI -- FROM INTERRUPT ROUTINE TO STORE STATUS IN THE CSW FOR ACTIVE PCI.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLC 001	T	360		CHANNEL INITIAL SELECTION	H.E.BERKEBILE
		DCLC 003	*				COUNT FIELD, HIGHORDERBYTE=UO
		DCLC 004	*				COUNT FIELD, LOW ORDER BYTE=U1
		DCLC 005	*				DATA ADDRESS, HIGH ORDER BYRE=VO
		DCLC 006	*				DATA ADDRESS, LOW ORDER BYTE =V1
		DCLC 007	*				(CMD)=GO
		DCLC 008	*				FLAGS AND OP BYTE =G1
		DCLC 009	*				TIMEOUT COUNTER =DO
		DCLC 010	*				DEVICE ADDRESS =D1
		DCLC 011	*				G1=0001 0000--- I/O INTERRUPT
		DCLC 012	*				G1=0000 0000--- TEST I/O
		DCLC 013	ASEQ	AL07=6E			
0AEE	C176	DCLC 014	ENTRY	018	NOPCIB	BR IF G1 BIT4=0	BR IF NO PCI BIT

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0AF0	7602	DCLC 015				STH D DA,88	STORE UNIT IDENT IN INT BUFFER
0AF2	2D40	DCLC 016				SET GB K=04	SET INT BUFFER LATCH
0AF4	8AF8	DCLC 017		019	ENTRY2	BR	
0AF6	0D40	DCLC 018	NOPCIB			RST GB K=04	RESET IB LTCH
0AF8	2B80	DCLC 019	ENTRY2			SET GA K=08	SET INL SELECT LATCH
0AFA	4F7F	DCLC 020	HLTENT			GB/OUT=D1	PUT ADR ON BUS OUT
0AFC	2B84	DCLC 021				SET GA K=28	SET ADDR-OUT
0AFE	2BC4	DCLC 022				SET GA K=2C	SET SEL-0 AND ADR-0
0B00	C9DD	DCLC 023	ADR1	069	OPINUP	BR IF GT BIT4=1	BR IF OP IN UP
0B02	FDA1	DCLC 024		039	CUBUSY	BR IF GT BIT3=1	BR IF STA-IN UP
0B04	262B	DCLC 025				DO=DO+K02	INCREMENT COUNTER
0B06	E600	DCLC 026		023	ADR1	BR IF DO BIT2=0	BR IF NO TIMEOUT
0B08	05C3	DCLC 027	SLIN			Z=G1*-K0C	MASK G1 BITS6 AND 7
0B0A	FC94	DCLC 028		033	ADR28	BR IF LZNZ	BR IF START I/O
0B0C	F514	DCLC 029		033	ADR28	BR IF G1 BIT3=0	BR IF TEST I/O
0B0E	2323	DCLC 030				V1=0\$K02	SET INTFCE CTRL CHECK
0B10	2207	DCLC 031				VO=0	SET 00 UNIT STATUS
0B12	8B3A	DCLC 032		052	ADR33C	BR	
0B14	DD9F	DCLC 033	ADR28	038	INFCHK	BR IF GT BIT1=1	BR IF SEL IN DOWN
0B16	2B00	DCLC 034	ADR29			SET GA K=00	RESET ALL TAGS OUT
0B18	F21F	DCLC 035		038	INFCHK	BR IF DO BIT7=1	BR IF CHAINED-TO CCW
0B1A	C3CB	DCLC 036		060	IPL002	BR IF BA BIT4=1	BR IF IPL
0B1C	AAF4	DCLC 037	ADR30	CCOM 165	CC3B	BR	
0B1E	A254	DCLC 038	INFCHK	DCLR 117	INFCHK	BR	
0B20	5F4F	DCLC 039	CUBUSY			GO=GB/IN	BUSS IN TO GO
0B22	FC9F	DCLC 040		038	INFCHK	BR IF GS BIT3=1	BR IF PARITY ERROR
0B24	2B84	DCLC 041				SET GA K=28	DROP SEL-OUT,KEEP ADR-0&INL SEL
0B26	E61F	DCLC 042	ADR10	038	INFCHK	BR IF DO BIT2=1	BR IF TIMEOUT
0B28	FDAE	DCLC 043		046	ADR9	BR IF GT BIT3=0	BR IF STA-IN DOWN
0B2A	262B	DCLC 044				DO=DO+K02	INCREMENT COONTER
0B2C	FDA7	DCLC 045		042	ADR10	BR IF GT BIT3=1	BR IF STA-IN STILL UP
0B2E	2B80	DCLC 046	ADR9			SET GA K=08	DROP ADR-OUT ,KEEP INL SEL ON
0B30	F21F	DCLC 047		038	INFCHK	BR IF DO BIT7=1	BR IF CHAINED-TO CCW
0B32	E149	DCLC 048		059	STRTIO	BR IF G1 BIT6=1	BR IF
0B34	F149	DCLC 049		059	STRTIO	BR IF G1 BIT7=1	START I/O
0B36	5429	DCLC 050	ADR33			VO=GO	MOVE UNIT STATUS TO VO
0B38	2307	DCLC 051	ADR33B			V1=0	SET 00 CHANNEL STATUS
0B3A	72A2	DCLC 052	ADR33C			STH V DA,AC	SAVE STATUS
0B3C	2B00	DCLC 053				SET GA K=00	DROP ALL TAGS OUT
0B3E	2400	DCLC 054	ADR34			SET MODE K=00	
0B40	5EA2	DCLC 055				RDH H DA,AC	RESTORE CHANNEL STATUS
0B42	2A07	DCLC 056				TO=0	
0B44	A01E	DCLC 057		CCOM 123	STATOS	BAL	BAL TO RTHE TO STORE CSW
0B46	AF44	DCLC 058		DCLT 041	INTRTX	BR	RTN FROM BAL FOR I/O INT ONLY
0B48	C308	DCLC 059	STRTIO	067	ADREBR	BR IF BA BIT4=0	BR IF NOT IPL
0B4A	875A	DCLC 060	IPL002	DCLL 021	IPLSTP	BR	GO TO IPL STOP
0B4C	F537	DCLC 061	ADR32	050	ADR33	BR IF G1 BIT3=1	BR IF I/O INTERRUPT
0B4E	5449	DCLC 062				GO=GO	
0B50	C486	DCLC 063		050	ADR33	BR IF ZNZ	BR IF STA NZ
0B52	2B00	DCLC 064				SET GA K=00	RESET INL SEL
0B54	B362	DCLC 065	ADR33A	CCOM 154	CC0B	BR	SET CDN CODE 0 & RTN TO ICYC
0B56	2B84	DCLC 066	HALTIO			SET GA K=28	RAISE ADR-0 AND DROPSEL-0
0B58	2307	DCLC 067	ADREBR			V1=0	CHNL STATUS = 0

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0B5A	A78A	DCLC 068		154	ADRE	BR	GO SET I/O INTERRUPT
0B5C	28C0	DCLC 069	OPIINUP			SET GA K=0C	RESET ADR-O
0B5E	0D10	DCLC 070				RST GB K=01	RST SUPR OUT
0B60	E61F	DCLC 071	ADR2	038	INFCHK	BR IF DO BIT2=1	BR IF TIMEOUT
0B62	CDE9	DCLC 072		075	ADINUP	BR IF GT BIT0=1	BR IF ADR-IN UP
0B64	262B	DCLC 073				DO=DO+K02	INCREMENT COONTER
0B66	CDE0	DCLC 074		071	ADR2	BR IF GT BIT0=0	BR IF ADR-IN NOT UP Y
0B68	4F4F	DCLC 075	ADINUP			GB/OUT=GO	PUT CMD ON BUSS OUT
0B6A	5F4F	DCLC 076				GO=GB/IN	PIT DEVICE ADR IN GO
0B6C	FC9F	DCLC 077		038	INFCHK	BR IF GS BIT3=1	BR IF PARITY ERROR
0B6E	6471	DCLC 078				GO=GO□D1	COMPARE ADDRESS
0B70	C49E	DCLC 079		038	INFCHK	BR IF ZNZ	BR IFADR MISMATCH
0B72	054B	DCLC 080				Z=G1□K04	TEST FOR HALT I/O
0B74	C4D7	DCLC 081		066	HALTIO	BR IF Z=0	BR IF HLT I/O
0B76	28C2	DCLC 082				SET GA K=1C	SET CMD-O
0B78	9E06	DCLC 083		085	ADR3	BR	
		DCLC 084	AEND				
1E06	E605	DCLC 085	ADR3	112	INFCKK	BR IF DO BIT2=1	BR IF TIMEOUT
1E08	FD8F	DCLC 086		089	STINUP	BR IF GT BIT3=1	BR IF STA-IN UP
1E0A	262B	DCLC 087				DO=DO+K02	INCREMENT COONTER
1E0C	FD86	DCLC 088		085	ADR3	BR IF GT BIT3=0	BR IF STA-IN NOT UP YET
1E0E	5F4F	DCLC 089	STINUP			GO=GB/IN	GO=STATUS ON BUSS-IN
1E10	FC85	DCLC 090		112	INFCKK	BR IF GS BIT3=1	BR IF PARITY ERROR
1E12	E123	DCLC 091		099	NOTTST	BR IF G1 BIT6=1	BR IF
1E14	F123	DCLC 092		099	NOTTST	BR IF G1 BIT7=1	START I/O
1E16	2B88	DCLC 093	TSTIS			SET GA K=48	SET SRV-OUT, INLSELUP
1E18	E6C5	DCLC 094	ADR31	112	INFCKK	BR IF DO BIT2=1	BR IF TIMEOUT
1E1A	C9A0	DCLC 095		098	ADR32B	BR IF GT BIT4=0	BR IF OP-IN DOWN
1E1C	262B	DCLC 096				DO=DO+K02	INCREMENT TMOUT CNTR
1E1E	C999	DCLC 097		094	ADR31	BR IF GT BIT4=1	BR IF OP-IN STILL UP
1E20	8B4C	DCLC 098	ADR32B	061	ADR32	BR	
1E22	C4BE	DCLC 099	NOTTST	114	NZSTA	BR IF ZNZ	BR IF NZ UNIT STATUS
1E24	F233	DCLC 100	ZEROST	107	NOBUFF	BR IF D07=1	BR IF CHAINED TO CCW
1E26	2DC4	DCLC 101				SET GB K=20	TURN ON BURST LATCH
1E28	3685	DCLC 102				DO=DO\$K80	TURN ON BURSTCH/FILE ACTIVE BIT
1E2A	7632	DCLC 103				STH D DA,8E	STORE STATS
1E2C	1685	DCLC 104				DO=DO*-K80	RESTORE DO TO ORIGINAL VALUE
1E2E	D134	DCLC 105		108	NOTBUF	BR IF G1 BIT5=0	BR IF NOT BUFFERED DEVICE
1E30	2D02	DCLC 106				SET GB K=10	TURN ON BUF DEV LTCH
1E32	1543	DCLC 107	NOBUFF			G1=G1*-K04	RESET BUFFERED DEV INDICATOR
1E34	2B48	DCLC 108	NOTBUF			SET GA K=44	SET SRV-OUT & KEEP SEL-OUT UP
1E36	1635	DCLC 109				DO=DO*-K30	RST 2 HI BITS OF TMOUT CNTR
1E38	C3BD	DCLC 110	IPLBR	113	IPL001	BR IF BA BIT4=1	BR IF IPL
1E3A	A7C0	DCLC 111		189	ADR24	BR	
1E04	A254	DCLC 112	INFCKK	DCLR	117	INFCHK	BR
1E3C	8AD0	DCLC 113	IPL001	DCLL	019	IPLWTL	BR
1E3E	C050	DCLC 114	NZSTA	123	NOCHE	BR IF GO BIT4=0	BR IF NO CH.E. BIT
1E40	F451	DCLC 115		123	NOCHE	BR IF GO BIT3=1	BR IF BSY BIT IN STAT
1E42	2D04	DCLC 116				SET GB K=20	TURN ON BURST LATCH
1E44	D14A	DCLC 117		120	NTBFX	BR IF G1 BIT5=0	BR IF NOT BUFFERED DEVICE
1E46	1543	DCLC 118				G1=G1*-K04	RESET BUFFERED DEVICE INDICATOR
1E48	2D02	DCLC 119				SET GB K=10	SET BUFFERED DEVICE LATCH
1E4A	2285	DCLC 120	NTBFX			VO=0\$K80	TURN ON BURSTCH/FILE ACTIVE BIT

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1E4C	2307	DCLC 121				V1=0	
1E4E	7232	DCLC 122				STH V DA,8E	STORE STATUS
1E50	2307	DCLC 123	NOCHE			V1=0	CHNL STATUS = 0
1E52	C561	DCLC 124		131	CCFLON	BR IF G1 BIT0=1	BR IF CDA FLG ON
1E54	D562	DCLC 125	ADRABC	132	ADR11	BR IF G1 BIT1=0	BR IF CCFLG =0
1E56	04C7	DCLC 126				Z=GO*-KCC	
1E58	C4E2	DCLC 127		132	ADR11	BR IF ZNZ	BR IF BAD STATUS
1E5A	C463	DCLC 128		132	ADR11	BR IF GO BIT0=1	BR IF BAD STATUS
1E5C	D065	DCLC 129		138	DECHTP	BR IF GO BIT5=1	BR IF DEVICE END
1E5E	CC7B	DCLC 130		133	ADR12	BR IF GO BIT4=1	BR IF CHNL END
1E60	15C5	DCLC 131	CCFLON			G1=G1*-KCO	CDA FLG ON - BRK CHAINING
1E62	A782	DCLC 132	ADR11	150	ADR11A	BR **TO NOT CHAINING OR BAD STATUS RTNE	
1E7A	2D10	DCLC 133	ADR12			SET GB K=01	SET SUPR OUT
1E7C	2B48	DCLC 134				SET GA K=44	SET SERVICE OUT
1E7E	2405	DCLC 135				GO=0\$K00	GO=00 CHNL STATUS
1E80	1573	DCLC 136				G1=G1*-K07	RST CODED OP BITS
1E82	9E38	DCLC 137		110	IPLBR	BR	
1E64	2D10	DCLC 138	DECHTP			SET GB K=01	SET SUPR OUT
1E66	2B88	DCLC 139				SET GA K=48	SET SRV-OUT
1E68	E6C5	DCLC 140	ADR15	112	INFCKK	BR IF DO BIT2=1	BR IF TIMEOUT
1E6A	C9F0	DCLC 141		144	ADR14	BR IF GT BIT4=0	BR IF OP IN DOWN
1E6C	262B	DCLC 142				DO=DO+K02	INCREMENT TMOU CNTR
1E6E	C9E9	DCLC 143		140	ADR15	BR IF GT BIT4=1	BR IF OP-IN STILL UP
1E70	2645	DCLC 144	ADR14			DO=0\$K40	RESET TMOU CNTR AND SET
1E72	3613	DCLC 145				DO=DO\$K01	UP DO TO CHAIN
1E74	5212	DCLC 146				RDH V DA,8A	RD OUT NXT CCW ADR
1E76	D402	DCLC 147		149	NOINCR	BR IF GO BIT1=0	BR IF NO STATUS MODIFIER
1E78	93D6	DCLC 148		DCLB 045	SKIP	BR	GO INCR CCW ADR+8 BEFORE CHAIN
1E02	93DA	DCLC 149	NOINCR	DCLB 054	RDHW1X	BR	NO CCW INCR NECESSARY-DO CHAIN
2782	C386	DCLC 150	ADR11A	152	NTIPL2	BR IF BA BIT4=0	BR IF NOT IPL
2784	8760	DCLC 151		DCLL 024	IPLSTB	BR	BR TO IPL STA RTNE
2786	F22B	DCLC 152	NTIPL2	169	NZSTCC	BR IF DO BIT7=1	BR IF CHAINED-TO CCW
2788	2B88	DCLC 153				SET GA K=48	SEND SRV-0
278A	5429	DCLC 154	ADRE			VO=GO	PUT UNIT STA IN VO
278C	C110	DCLC 155		157	ADR16	BR IF G1 BIT4=0	BR IF PCI FLG OFF
278E	3385	DCLC 156				V1=V1\$K80	PUT PCI BIT IN CH STA
2790	7222	DCLC 157	ADR16			STH V DA,8C	TEMP STORE STATUS
2792	2B00	DCLC 158				SET GA K=00	RESET INL SELECT
2794	2CC7	DCLC 159				UO=0	
2796	7032	DCLC 160				STH U DA,8E	AUX008E = 0000 --RST STATS
2798	0D04	DCLC 161				RST GB K=20	RESET BURST LATCH
279A	E629	DCLC 162	ADR18	174	INFCKX	BR IF DO BIT2=1	BR IF TIMEOUT
279C	C9A2	DCLC 163		166	ADR17	BR IF GT BIT4=0	BR IF OP-IN DOWN
279E	262B	DCLC 164				DO=DO+K02	INCREMENT TMOU COUNTER
27A0	C99B	DCLC 165		162	ADR18	BR IF GT BIT4=1	BR IF OP-IN STILL UP
27A2	2400	DCLC 166	ADR17			SET MODE K=00	
27A4	5E22	DCLC 167				RDH H DA,8C	RD OUT UNIT &CHNL STATUS
27A6	A030	DCLC 168		CCOM 135	STATUS	BR	BR TO STORE CSW ROUTINE
27AA	5429	DCLC 169	NZSTCC			VO=GO	XFER STATUS TO VO
27AC	5349	DCLC 170				GO=V1	GO = CHNL STATUS
27AE	ECC5	DCLC 171		175	ADR20	BR IF GS BIT2=1	BR IF BURST LTCH ON
27B0	D8C4	DCLC 172	TRYIBS	175	ADR20	BR IF GS BIT5=0	BR IF IBLTCH=0
27B2	A7B4	DCLC 173		183	ADR22	BR	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
27A8	A254	DCLC 174	INFCKX	DCLR 117	INFCHK	BR	
27C4	1635	DCLC 175	ADR20			DO=DO*-K30	
27C6	7602	DCLC 176				STH D DA,88	STORE UNIT IDENT HWD IN IB
27C8	3413	DCLC 177				GO=GO\$K01	TURN ON STATUS RECD B
27CA	2D40	DCLC 178				SET GB K=04	TURN ON IB LATCH
27CC	ECB4	DCLC 179		183	ADR22	BR IF GS BIT2=0	BR IF BRST LTCH OFF
27CE	2B08	DCLC 180				SET GA K=40	SRV-O---ACCEPT STATUS
27D0	3413	DCLC 181				GO=GO\$K01	TURN ON STA RECD BIT
27D2	A7B8	DCLC 182		185	ADR23	BR	
27B4	2B02	DCLC 183	ADR22			SET GA K=10	CMD-O---QUE STATUS
27B6	1635	DCLC 184	ADR22A			DO=DO*-K30	
27B8	E629	DCLC 185	ADR23	174	INFCKX	BR IF DO BIT2=1	BR IF TIMEOUT
27BA	C9C0	DCLC 186		189	ADR24	BR IF GT BIT4=0	BR IF OP IN DOWN
27BC	262B	DCLC 187				DO=DO+K02	INCREMENT TMOUT COUNTER
27BE	C9B9	DCLC 188		185	ADR23	BR IF GT BIT4=1	BR IF OP-IN STILL UP
27C0	E680	DCLC 189	ADR24	191	ADR25	BR IF MMSK2=0	BR IF NOT IN TRAP
27C2	0244	DCLC 190				RTN G MMSK2=0	RETURN FROM TRAP
2780	A162	DCLC 191	ADR25	DCHN 092	MODRST	BR	RTN TO I CYCLES
		DCLC 192	*				
		DCLC 193	ATABLE	ADDR=0084			
0084	0B0C	DCLC 194	C			XAUX'0B0C0D0E'	BUFFERED DEVICE STANDARD ADDRESS
		DCLC 195	AEND				
		DCLC 196	ATABLE	ADDR=0094			
0094	0F15	DCLC 197	C			XAUX'0F1518FF'	BUFFERED DEVICE STANDARD ADDRESS
		DCLC 198	AEND				

 * CROSS REFERENCE FOR CSECT DCLC *

DCLC 014	DCLB 159							
DCLC 018	DCLC 014	DCLH 015						
DCLC 019	DCLC 017							
DCLC 020	DCLH 014							
DCLC 023	DCLC 026							
DCLC 033	DCLC 028	DCLC 029						
DCLC 038	DCLC 033	DCLC 035	DCLC 040	DCLC 042	DCLC 047	DCLC 071	DCLC 077	DCLC 079
DCLC 039	DCLC 024							
DCLC 042	DCLC 045							
DCLC 046	DCLC 043							
DCLC 050	DCLC 061	DCLC 063						
DCLC 052	DCLC 032							
DCLC 059	DCLC 048	DCLC 049						
DCLC 060	DCLC 036							
DCLC 061	DCLC 058							
DCLC 066	DCLC 081							
DCLC 067	DCLC 059							
DCLC 069	DCLC 023							
DCLC 071	DCLC 074							
DCLC 075	DCLC 072							
DCLC 085	DCLC 083	DCLC 088						
DCLC 089	DCLC 086							
DCLC 094	DCLC 097							
DCLC 098	DCLC 095							
DCLC 099	DCLC 091	DCLC 092						

* CROSS REFERENCE FOR CSECT DCLC *

DCLC 107	DCLC 100			
DCLC 108	DCLC 105			
DCLC 110	DCLC 137			
DCLC 112	DCLC 085	DCLC 090	DCLC 094	DCLC 140
DCLC 113	DCLC 110			
DCLC 114	DCLC 099			
DCLC 120	DCLC 117			
DCLC 123	DCLC 114	DCLC 115		
DCLC 125	DCLC 073			
DCLC 131	DCLC 124			
DCLC 132	DCLC 125	DCLC 127	DCLC 128	
DCLC 133	DCLC 130			
DCLC 138	DCLC 129			
DCLC 140	DCLC 143			
DCLC 144	DCLC 141			
DCLC 149	DCLC 147			
DCLC 150	DCLC 132	DCLC 070		
DCLC 152	DCLC 150			
DCLC 154	DCLC 068			
DCLC 157	DCLC 155			
DCLC 162	DCLC 165			
DCLC 166	DCLC 163			
DCLC 169	DCLC 152			
DCLC 172	DCLC 055			
DCLC 174	DCLC 162	DCLC 185		
DCLC 175	DCLC 171	DCLC 172		
DCLC 183	DCLC 173	DCLC 179		
DCLC 184	DCLC 056			
DCLC 185	DCLC 182	DCLC 188		
DCLC 189	DCLC 111	DCLC 186		
DCLC 191	DCLC 189			

DCLD DESCRIPTIVE TEXT

CHANNEL DATA LOOP

THIS ROUTINE PROVIDES THE DATA LOOP FOR EITHER THE BYTE CHANNEL-0 OR BURST CHANNEL-1, DEPENDING ON WHICH CHANNEL OPTION IS LOADED IN CONTROL STORAGE.

BYTE CHANNEL ENTRY POINTS

ENTRY -- COMMON ENTRY INTO A LOOP THAT CHECKS FOR THE RISE OF SERVICE-IN, STATUS-IN, OR THE FALL OF OPERATIONAL-IN. THIS ENTRY INITIALIZES THE TIMEOUT COUNTER.

LOOP -- SAME AS -ENTRY- EXCEPT TIMEOUT COUNTER IS NOT INITIALIZED.

RESET -- FROM LOW-PRIORITY-TRAP ROUTINE -DCLE- OR CHANNEL ERROR ROUTINE -DCLR- TO RESET TAGS-OUT, THEN EITHER RETURN FROM TRAP OR GO TO I-CYCLES.

SERV -- FROM -DCLE- STATUS ROUTINE WHEN THE DEVICE REQUESTS SERVICE.

DECREM -- FROM -DCLR- CHANNEL ERROR ROUTINE AFTER CHANNEL STATUS IS SET FOR PROTECTION CHECK OR STORAGE WRAP CHECK. PURPOSE OF ENTRY IS TO DECREMENT COUNT, THEN RETURN FROM TRAP.

BURST CHANNEL ENTRY POINTS

0170 -- TRAP ENTRY FOR DATA SERVICE ON NON-BUFFERED DEVICE.

OPBRCH -- ENTRY FROM LOW-PRIORITY-TRAP ROUTINE -DCLE- FOR BUFFERED DEVICES.

DECCNT -- FROM -DCLR- ROUTINE AFTER CHANNEL STATUS IS SET FOR PROTECTION CHECK OR STORAGE WRAP CHECK. PURPOSE OF ENTRY IS TO DECREMENT COUNT, THEN RETURN FROM TRAP OR GO TO I-CYCLES.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLD 001	T			CHANNEL DATA LOOP	H.E.BERKEBILE
		DCLD 009	ATABLE	ADDR=0170			
0170	2240	DCLD 010	LABEL			LINK G MMSKO=1	FIRST WORD OF TRAP
0172	8153	DCLD 011	OPBRCH	013	OP	N N=G1 BITS67	BR ON OP
		DCLD 012	AEND				
01C0	81EC	DCLD 013	OP	0	035	EXPSTA BR	COUNT=0 AND STATUS EXPECTED
01C2	81DC	DCLD 014	OP	1	027	OUTPUT BR	OUTPUT COMMAND
01C4	F554	DCLD 015	OP	2	023	RDFWD BR IF G1 BIT3=0	BR IF SKIP FLAG IS 0
01C6	F551	DCLD 016	OP	3	021	SKIP BR IF G1 BIT3=1	BR IF SKIP FLAG IS 1
01C8	7F3E	DCLD 017				STB GB/IN V-1	STORE BUS IN,ADR -1
01CA	5006	DCLD 018	DECCNT			U=U-1	COUNT FIELD -1
01CC	C4E3	DCLD 019		030	CTZERO	BR IF Z=0	BR IF COUNT = 0
01CE	0240	DCLD 020				RTN G MMSKO=0	RETURN FORM TRAP
01D0	2B48	DCLD 021	SKIP			SET GA K=44	SET SRV-OUT
01D2	81CA	DCLD 022		018	DECCNT	BR	
01D4	7F3C	DCLD 023	RDFWD			STB GB/IN V+1	STORE BUS IN,ADR +1
01D6	5006	DCLD 024				U=U-1	DECREMENT COUNT BY 1
01D8	C4E3	DCLD 025		030	CTZERO	BR IF Z=0	BR IF COUNT HAS GONE TO ZERO
01DA	0240	DCLD 026	CTNZ			RTN G MMSKO=0	RETURN FORM TRAP
01DC	5F3C	DCLD 027	OUTPUT			RDB GB/OUT V+1	RD OUT TO BUS-0,ADR+1
01DE	5006	DCLD 028				U=U-1	COUNT -1
01E0	C4DA	DCLD 029		026	CTNZ	BR IF ZNZ	BR IF COUNT IS NOT YET ZERO

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
01E2	C569	DCLD 030	CTZERO	033	CTCDA	BR IF G1 BIT0=1	BR IF CD FLAG
01E4	1573	DCLD 031				G1=G1*-K07	SET CT=0 INDICATION
01E6	0240	DCLD 032				RTN G MMSKO=0	RETURN FORM TRAP
01E8	3D00	DCLD 033	CTCDA			SET GB K=80	SET CDA REQ
01EA	0240	DCLD 034				RTN G MMSKO=0	RETURN FORM TRAP
01EC	2842	DCLD 035	EXPSTA			SET GA K=14	SET CMD OUT
01EE	D169	DCLD 036		033	CTCDA	BR IF G1 BIT5=1	BR IF CMD 0 SET PREV
01F0	3543	DCLD 037				G1=G1\$K04	G1 BIT5=1 CMD=0 GIVE
01F2	0240	DCLD 038				RTN G MMSKO=0	RETURN FORM TRAP

 * CROSS REFERENCE FOR CSECT DCLD *

DCLD 011	DCLD 031	
DCLD 013	DCLD 011	
DCLD 018	DCLD 022	DCLR 112
DCLD 021	DCLD 016	
DCLD 023	DCLD 015	
DCLD 026	DCLD 029	
DCLD 027	DCLD 014	
DCLD 030	DCLD 019	DCLD 025
DCLD 033	DCLD 030	DCLD 036
DCLD 035	DCLD 013	

DCLE DESCRIPTIVE TEXT

CHANNEL LOW PRIORITY TRAP STATUS ROUTINES

BYTE CHANNEL

WHEN THE BYTE CHANNEL OPTION IS LOADED IN CONTROL STORAGE, THIS ROUTINE HANDLES ALL TRAPS -- EITHER DATA SERVICE OR STATUS.

BYTE CHANNEL ENTRY POINTS

0180 -- LOW-PRIORITY TRAP-ADDRESS FOR DATA OR STATUS.

STATUS -- FROM ENTRY LOOP OF -DCLD- WHEN STATUS-IN RISES. THIS PORTION OF -DCLE- HANDLES STATUS.

BURST CHANNEL

WHEN THE BURST CHANNEL OPTION IS LOADED IN CONTROL STORAGE, THIS ROUTINE HANDLES TRAPS FOR EITHER --
1. DATA SERVICE - BUFFERED DEVICE, OR
2. STATUS - ANY DEVICE.

BURST CHANNEL ENTRY POINTS

0180 -- LOW-PRIORITY TRAP-ADDRESS AS DESCRIBED ABOVE.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLE 001	T			CHANNEL LOW PRIORITY TRAP STATUS ROUTINES	H.E.BERKEBILE
		DCLE 009	ATABLE	ADDR=0180			
0180	2244	DCLE 010				LINK G MMSK2=1	TRAP WORD-SET PRIORITY
0182	3613	DCLE 011				DO=D0\$K01	SET SOURCE OF ENTRY INDICATOR
0184	CCA4	DCLE 012		028	ADR40	BR IF GS BIT0=0	BR IF CDA REQ=0
0186	D123	DCLE 013		027	ADR41	BR IF G1 BIT5=1	BR IF ICC DETECTED
0188	5212	DCLE 014				RDH V DA,8A	READ OUT NEXT CCW ADR
018A	5038	DCLE 015				RDH U V+2	READ CCW HWD 1
018C	008B	DCLE 016				Z=U0\$K08	TEST COMMAND FOR TIC
018E	F0A0	DCLE 017		026	NOTIC	BR IF LZNZ	BR IF NOT A TIC
0190	5119	DCLE 018				U1=U1	
0192	C49C	DCLE 019		024	PRGERD	BR IF ZNZ	BR IF TIC ADDR TOO BIG
0194	5230	DCLE 020				RDH V V+0	RD OUT NEW CCW ADDR
0196	0383	DCLE 021				Z=V1*-K08	CHECK FOR DOUBLE WORD BOUNDARY
0198	FC9C	DCLE 022		024	PRGERD	BR IF LZNZ	
019A	93F4	DCLE 023		DCLB 077	RDHW1	BR	
019C	5212	DCLE 024	PRGERD			RDH V DA,8A	READ OUT NEXT CCW ADR
019E	93C2	DCLE 025		DCLB 106	PRGERC	BR	INCR CCW ADR IN DCLB & BR TO DCLR
01A0	93FA	DCLE 026	NOTIC	DCLB 080	NOTIC	BR	BR TO DCLB & CONTINUE CCW FETCH
01A2	A254	DCLE 027	ADR41	DCLR 117	INFCHK	BR	BR TO INTFCE CTRL CHECK RTNE
01A4	9A34	DCLE 028	ADR40	030	ADR40B	BR	
		DCLE 029	AEND				
1A34	EDB0	DCLE 030	ADR40B	032	ADR40A	BR IF GT BIT2=0	BR IF NO SRV-IN
1A36	8172	DCLE 031		DCLD 011	OPBRCH	BR	
1A30	FDB8	DCLE 032	ADR40A	034	NTSTIN	BR IF GT BIT3=0	BR IF STA-IN=0
1A32	929E	DCLE 033		058	CHESTA	BR	STATUS IS CHANNEL END TYPE
1A38	2B40	DCLE 034	NTSTIN			SET GA K=04	SET SEL OUT
1A3A	2645	DCLE 035				DO=0\$K40	SET DO
1A3C	3613	DCLE 036				DO=D0\$K01	TO 41

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1A3E	92B8	DCLE 037		038	ADR43	BR	
1288	CD89	DCLE 038	ADR43	046	ADR44	BR IF GT BIT0=1	BR IF ADR IN UP
12BA	262B	DCLE 039				DO=DO+K02	INCREMENT TMOUT COUNTER
128C	CD89	DCLE 040		046	ADR44	BR IF GT BIT0=1	BR IF ADR IN UP
12BE	E638	DCLE 041		038	ADR43	BR IF DO BIT2=0	BR IF NO TIMEOUT
12C0	DD9D	DCLE 042		045	INFCHK	BR IF GT BIT1=1	BR IF NO SEL-IN
12C2	2B00	DCLE 043				SET GA K=00	SEL-IN RECEIVED-DROP ALL TAGS DU
12C4	0244	DCLE 044				RTN G MMSK2=0	AND RETURN FROM TRAP
129C	A254	DCLE 045	INFCHK	DCLR 117	INFCHK	BR	BR TO ERROR RTNES FOR ICC
1288	5F7F	DCLE 046	ADR44			D1=GB/IN	D1=ADR ON BUSS-IN
128A	FC9D	DCLE 047		045	INFCHK	BR IF GS BIT3=1	BR IF BAD PARITY ON BUSS-IN
128C	2007	DCLE 048				U0=0	
128E	4F0F	DCLE 049				GB/DUT=U0	SET BUSS-OUT TO ZERO
1290	2B02	DCLE 050				SET GA K=10	SET CMD-D---PROCEED
1292	E61D	DCLE 051	ADR47	045	INFCHK	BR IF DO BIT2=1	BR IF TIMEOUT
1294	FD9B	DCLE 052		055	ADR46	BR IF GT BIT3=1	BR IF STA-IN UP
1296	262B	DCLE 053				DO=DO+K02	INCREMENT TIMEOUT COUNTER
1298	FD92	DCLE 054		051	ADR47	BR IF GT BIT3=0	BR IF STA-IN DWN
129A	A780	DCLE 055	ADR46	DCLC 172	TRYIBS	BR	TRY TO LOAD INT BUFFER
		DCLE 056	*				**SET IB AND QUE
		DCLE 057	*				**STATUS AT DEVICE
129E	FCA4	DCLE 058	CHESTA	061	ADR48	BR IF GS BIT3=0	BR IF DATA PARITY OK
12A0	0DC2	DCLE 059				RST GB K=10	RESET PARITY CHECK LATCH
12A2	3483	DCLE 060				GO=GO\$K08	PUT CDC BITIN CHNLSTA
12A4	5439	DCLE 061	ADR48			V1=GO	MUVE CHANNEL STATUS
12A6	C52A	DCLE 062		064	NCCRST	BR IF G1 BIT0=0	BR IF NO CDA FLG
12A8	1565	DCLE 063				G1=G1*-K60	ZERO OUT CC AND SLI FLGS
12AA	5F4F	DCLE 064	NCCRST			GO=GB/IN	PUT GB/IN INTO GO
12AC	FC9D	DCLE 065		045	INFCHK	BR IF GS BIT3=1	BR IF PARITY ERROR
12AE	E503	DCLE 066		071	ADR50	BR IF G1 BIT2=1	BR IF SLI BIT=1
12B0	0583	DCLE 067				Z=G1*-K08	MASK G1-5,6,7
12B2	FC83	DCLE 068		071	ADR50	BR IF LZ=0	BR IF CLR
12B4	3345	DCLE 069				V1=V1\$K40	PUTINCLNGTHINCHNLSTA
12B6	A782	DCLE 070	ADR51	DCLC 150	ADR11A	BR	TRY TO SET INTRUPT BUFFER
1282	0313	DCLE 071	ADR50			Z=V1*-K01	MASK CHANNEL STATUS
1284	C486	DCLE 072		070	ADR51	BR IF ZNZ	BR IF CHNL STATUS NZ
1286	9E54	DCLE 073		DCLC 125	ADRABC	BR	CONTINUE TESTING STATUS

 * CROSS REFERENCE FOR CSECT DCLE *

DCLE 024	DCLE 019	DCLE 022			
DCLE 026	DCLE 017				
DCLE 027	DCLE 013				
DCLE 028	DCLE 012				
DCLE 030	DCLE 028				
DCLE 032	DCLE 030				
DCLE 034	DCLE 032				
DCLE 038	DCLE 037	DCLE 041			
DCLE 045	DCLE 042	DCLE 047	DCLE 051	DCLE 065	
DCLE 046	DCLE 038	DCLE 040			
DCLE 051	DCLE 054				
DCLE 055	DCLE 052				
DCLE 058	DCLE 033				

* CROSS REFERENCE FOR CSECT DCLE *

DCLE 061	DCLE 058
DCLE 064	DCLE 062
DCLE 070	DCLE 072
DCLE 071	DCLE 066 DCLE 068

DCLH DESCRIPTIVE TEXT

HALT I/O ROUTINE

THIS ROUTINE EXECUTES A HALT I/O INSTRUCTION WHEN THE BURST CHANNEL OPTION IS LOADED IN CONTROL STORAGE.

ENTRY POINTS

ENTRY -- FROM -DCLA- I/O INSTRUCTIONS ROUTINE WHEN A HALT I/O INSTRUCTION IS DECODED.

ENTRY POINTS CONTINUED

ADRO -- FROM -DCLT- FOR INACTIVE TEST I/O TO TRANSFER INFORMATION FROM ONE LOCAL STORAGE ZONE TO ANOTHER.

ADROA -- FROM -DCLT- FOR INACTIVE I/O INTERRUPT TO TRANSFER INFORMATION FROM ONE LOCAL STORAGE ZONE TO ANOTHER.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLH 001	T		CHANNEL	HALT I/O	H.E.BERKEBILE
		DCLH 002	*				
2340	C625	DCLH 004	ENTRY	016	ADR1	BR IF DO BIT0=1	BR IF ACTIVE
2342	2043	DCLH 005				UO=0\$K04	SET UP UOTO INDICATE HLT I/O
2344	7022	DCLH 006	ADRO			STH U DA,8C	TEMP STORE UNIT IDENT HWD
2346	24F6	DCLH 007	ADROA			SET MODE K=3F	SET CHNL MODE & ZONE
2348	2B80	DCLH 008				SET GA K=08	SET INL SELECT LTCH
234A	5622	DCLH 009				RDH D DA,8C	RD OUT UNIT IDENT
234C	5659	DCLH 010				G1=DO	G1 = IST I/O OR HLT I/O IDENT
234E	2645	DCLH 011				DO=0\$K40	SET D TO IDENTIFY BURST CHNL
2350	2407	DCLH 012				GO=0	GO = CHNL STATUS
2352	D122	DCLH 013		015	HLTEBR	BR IF G1 BIT5=0	BR IF HLT I/O
2354	8AFA	DCLH 014		DCLC 020	HLTENT	BR	GO GET DEVICE OM INTERFACE
2322	8AF6	DCLH 015	HLTEBR	DCLC 018	NOPCIB	BR	GO GET DEVICE ON INTERFACE
2324	2C25	DCLH 016	ADR1			PO=0\$K20	SET CC2
2326	24F6	DCLH 017				SET MODE K=3F	SET CHNL MODE
2328	2210	DCLH 018				SET MMSK K=01	SET DATA-TRAP PRIORITY
232A	F030	DCLH 019		022	ADR6	BR IF GO BIT7=0	BR IF NO STA RECD
232C	0210	DCLH 020				RST MMSK K=01	RST DATA-TRAP PRIORITY
232E	B362	DCLH 021		CCOM 154	CCOB	BR	BR TO CCO SET RTNE
2330	1D00	DCLH 022	ADR6			RST GB K=80	RST CDA REQ LTCH
2332	2B84	DCLH 023				SET GA K=28	GIVE SELECTIVE RESET
2334	0210	DCLH 024				RST MMSK K=01	RST DATA-TRAP PRIORITY
2336	E53F	DCLH 025		029	NOINCL	BR IF G1 BIT2=1	BR IF SLI BIT ON
2338	4006	DCLH 026				U=U	U=COUNT
233A	C4BF	DCLH 027		029	NOINCL	BR IF Z=0	BR IF COUNT = 0
233C	3445	DCLH 028				GO=GO\$K40	PUT INC LGTH BIT IN CHNL STATS
233E	B0B0	DCLH 029	NOINCL	DCLR 048	PRGECC	BR	BR TO RTNE TO SET INT BUFFER

 * CROSS REFERENCE FOR CSECT DCLH *

DCLH 004 DCLA 111
 DCLH 006 DCLT 005
 DCLH 007 DCLT 060
 DCLH 015 DCLH 013

* CROSS REFERENCE FOR CSECT DCLH *

DCLH 016 DCLH 004
DCLH 022 DCLH 019
DCLH 029 DCLH 025 DCLH 027

DCLL DESCRIPTIVE TEXT

ENTRY POINTS

IPL - ENTRY IS MADE TO THIS LABEL FROM THE -BSYS- ROUTINE. THE FOLLOWING STEPS HAVE TAKEN PLACE PRIOR TO ENTRY HERE -

1. THE LOAD KEY HAS BEEN PRESSED AND A MACHINE RESET PERFORMED.
2. THE RESIDENT DIAGNOSTIC -BDIA- WAS EXECUTED.
3. THE CHECK SUM ROUTINE -BCHK- HAS BEEN EXECUTED.
4. THE SYSTEM RESET ROUTINE -BSYS- HAS BEEN EXECUTED AND A BRANCH TO THIS ROUTINE AT LABEL * IPL * HAS BEEN PERFORMED.

IPLWTL - IF OPERATING AS (BURST CHANNEL), ENTRY IS MADE AT THIS LABEL FROM THE -DCLC- ROUTINE TO ENTER THE IPL WAIT LOOP AND WAIT FOR A TRAP TO ADDRESS 0170 IN THE -DCLD- ROUTINE, OR, IF BUFFERED DEVICE, TRAP TO ADDRESS 0180 IN THE -DCLE- ROUTINE.

IF OPERATING AS (BYTE CHANNEL), ENTRY IS MADE AT THIS LABEL FROM THE -DCLD- ROUTINE AFTER OPERATIONAL-IN HAS DROPPED. THE WAIT LOOP IS ENTERED TO WAIT FOR A TRAP TO ADDRESS 0180 IN THE -DCLE- ROUTINE.

IPLSTP - ENTRY IS MADE TO THIS LABEL BECAUSE OF SOME UNRECOVERABLE ERROR BEING DETECTED. A HARD STOP IS EXECUTED.

IPLSTB - ENTRY IS MADE TO THIS LABEL IF OPERATING (BURST CHANNEL), FROM THE -DCLC- ROUTINE TO CHECK CHANNEL STATUS AND THE COMMAND CHAINING FLAG.

IPLSTA - ENTRY IS MADE TO THIS LABEL IF OPERATING (BYTE CHANNEL), FROM THE -DCLC- ROUTINE AFTER ALL CCWS HAVE BEEN READ, TO STORE ZEROS IN THE FLAGS AND OP HALFWORD OF THE UCW.

PICKUP - ENTRY IS MADE TO THIS LABEL FROM EITHER THE 2540 REQUEST AND TEST I/O ROUTINE -ERRQ-, OR THE 2311 FEATURE, COMMAND VERIFICATION AND EXECUTION ROUTINE -FILX-. ENTRY IS MADE AFTER THE IPL IS COMPLETED FROM THE DEVICE TO STORE THE DEVICE ADDRESS IN PROGRAM STORAGE LOCATION 0002 AND BRANCH TO THE -BSYS- ROUTINE TO RESET IPL AND GO TO -BPSW- TO LOAD THE PSW.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLL 001	T			INITIAL PROGRAM LOAD (IPL)	H.E.BERKEBILE
25DA	42A6	DCLL 002	IPL			V=T	T AND U ARE BOTH ZERO-FROM SYSR
25DC	44A6	DCLL 003				G=T	
25DE	2423	DCLL 004				GO=0\$K02	G=0200 (READ CMD)
25E0	74B8	DCLL 005				STH G T+2	MS--0000=0200
25E2	70B8	DCLL 006				STH U T+2	MS--0002=0000
25E4	2465	DCLL 007				GO=0\$K60	G=6000 (CC&SLI FLGS)
25E6	74B8	DCLL 008				STH G T+2	MS---0004=6000
25E8	2115	DCLL 009				U1=0\$K10	
25EA	3183	DCLL 010				U1=U1\$K08	U=C018 (COUNT)
25EC	70B8	DCLL 011				STH U T+2	MS---0006=0018
25EE	3B45	DCLL 012				T1=T1\$K40	T=C048 FOR UNIT ADDRESS IN DCLA
25F0	46A6	DCLL 013				D=T	D=C048
25F2	17FD	DCLL 014				D1=D1\$KFO	D=C0B8
25F4	8ADA	DCLL 015		CSTS 066	RDSWCH	BAL	READ IPL ADDR
25F6	10F5	DCLL 016				U0=U0*-KFO	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
25F8	24C3	DCLL 017				GO=0\$K0C	
25FA	87AE	DCLL 018		DCLA 089	UADRID	BR	
UAD0	0214	DCLL 019	IPLWTL			RST MMSK K=21	RESET MMSK 2
UAD2	8AD2	DCLL 020	WAITLP	020	WAITLP	BR	IPL WIAT LOOP
075A	3B00	DCLL 021	IPLSTP			SET GA K=80	DRUP OP-OUT
075C	5C07	DCLL 022	STOP			STOP	IPL STOP
075E	875C	DCLL 023		022	STOP	BR	GO BACK TO STOP AGAIN
0760	0313	DCLL 024	IPLSTB			Z=V1*-K01	
0762	C4DA	DCLL 025		021	IPLSTP	BR IF ZNZ	BR IF BAD CHNL STATUS
0764	04C3	DCLL 027				Z=GO*-K0C	
0766	C4DA	DCLL 028		021	IPLSTP	BR IF ZNZ	BR IF BAD STATUS
0768	D55B	DCLL 029		021	IPLSTP	BR IF G1 BIT1=1	BR IF NO CC FLAG
076A	0D04	DCLL 030				RST GB K=20	RST BURST LTCH
076C	2407	DCLL 031				GO=0	
076E	7432	DCLL 032				STH G DA,8E	RESET STATS IN AUX008E
0770	2B08	DCLL 033				SET GA K=40	SET SRV=0
0772	C9F3	DCLL 034	OPINLP	034	OPINLP	BR IF GT BIT4=1	BR ON FALL OF OP-IN
0774	0214	DCLL 035				RST MMSK K=21	RST MMSK 2
0776	2400	DCLL 036	PICKUP			SET MODE K=00	
0778	8ADA	DCLL 037		CSTS 066	RDSWCH	BAL	READ IPL ADDR
077A	10F5	DCLL 038				UO=UO*-KFO	
077C	2B23	DCLL 039				T1=0\$K02	
077E	70B0	DCLL 040				STH U T+0	
0780	A1CA	DCLL 041		BSYS 125	RETURN	BR	GO RST IPL AND LOAD PSW

 * CROSS REFERENCE FOR CSECT DCLL *

DCLL 002	BSYS 123							
DCLL 019	DCLC 113							
DCLL 020	DCLL 020							
DCLL 021	DCHN 095	DCLC 060	DCLL 025	DCLL 028	DCLL 029	DCLR 003	DCLR 116	
DCLL 022	DCLL 023							
DCLL 024	DCLC 151							
DCLL 034	DCLL 034							
DCLL 036	ERRQ 047	FILX 428						

DCLR DESCRIPTIVE TEXT

CHANNEL ERROR ROUTINES

ENTRY POINTS CONTINUED

ENTRY POINTS

PRGERR -- BURST CHANNEL ENTRY FROM -DCLB- WHENEVER A PROGRAM ERROR IS DETECTED WHILE FETCHING A CCW FOR A START I/O OR CHAINING.

PGESTT -- FROM -DCLA- WHEN A PROGRAM ERROR IS DETECTED WHILE FETCHING THE CAW FOR START I/O.

NTYPE -- FROM -BWRP- WHEN A STORAGE WRAP OR STORAGE PROTECT VIOLATION IS DETECTED WITH MODE SET FOR ONE OF THE INTEGRATED ATTACHMENTS.

PRGECC -- THIS BURST CHANNEL ENTRY LOADS THE INTERRUPT BUFFER, THEN THE MICROPROGRAM EITHER RETURNS FROM A TRAP OR GOES TO I-CYCLES. ENTRY IS FROM ONE OF THE FOLLOWING.
 1. -DCLR- IF PROGRAM ERROR IS DETECTED WHILE COMMAND CHAINING.
 2. -DCLH- AFTER PERFORMING HALT I/O.

INTQD -- BYTE-CHANNEL ENTRY FROM -DCLC- WHEN THE INTERRUPT BUFFER IS FULL AND ANY CHANNEL ERROR OCCURS. THE DEVICE IS SELECTED AND THE STATUS IS QUEUED IN -DCLC-. IN -DCLR-, THE UCW IS UPDATED, THEN THE ROUTINE BRANCHES TO -DCLD- TO EITHER RETURN TO I-CYCLES OR TO RETURN FROM TRAP.

WRAP
 PROTCH -- FROM -BWRP- TO SET EITHER PROGRAM CHECK OR PROTECTION CHECK IN STATUS, AND SET UP TO STOP THE DEVICE ON THE NEXT SERVICE-IN.

INFCHK -- COMMON ENTRY FOR INTERFACE CONTROL CHECK LOG-OUT. THE INTERFACE IS CLEARED AND THE CHANNEL ERROR STATUS IS SET UP. THEN THE ROUTINE EITHER RETURNS FROM THE TRAP OR GOES TO I-CYCLES, DEPENDING ON SOURCE OF ENTRY.

CCCXXX -- FROM -BMCK- AFTER MACHINE CHECK TRAP IF MMSK BITS INDICATE A CHANNEL CONTROL CHECK.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLR 001	T			CHANNEL ERROR ROUTINES	H.E. BERKEBILE
3082	C3C2	DCLR 002	PRGERR	004	NOIPLM	BR IF BA4=0	BR IF NOT IPL
3084	875A	DCLR 003		DCLL 021	IPLSTP	BR	GO TO IPL STOP
30C2	F207	DCLR 004	NOIPLM	013	PGECHN	BR IF D07=1	BR IF FROM CHAINING
30C4	2400	DCLR 008	PGESTT			SET MODE K=00	
30C6	2F25	DCLR 009				H1=0\$K20	SET PROGRAM CHECK IN CHNL STATUS
30C8	2E05	DCLR 010	ADRO03			H0=0\$K00	H0=00 UNIT STA H1=CHNL STA
30CA	A030	DCLR 011		CCCM 135	STATUS	BR	STORE STATUS ONLY CSW-SET CCI
3086	D623	DCLR 013	PGECHN	041	BURST	BR IF D0 BIT1=1	BR IF BURSTCH
3088	1773	DCLR 017				D1=D1*-K07	SET D TO UCW HWD 0
308A	AC78	DCLR 028	NOTHEM	ERRQ 083	RDCHST	BAL	GET CHAN AND UNIT STATUS IN G,H
308C	C510	DCLR 029		031	NODATC	BR IF G10=0	BR IF NO DATA CHAIN
308E	2EC3	DCLR 030				H0=0\$K0C	SET CHANNEL END AND DEVEND
3090	15C5	DCLR 031	NODATC			G1=G1*-K0C	RESET CHAIN FLAGS
3092	34A5	DCLR 032				G0=G0\$KAO	SET SEC AND PROG CHECK
3094	C398	DCLR 033	NOCOMM	035	NATPRO	BR IF BA4=0	BR IF NOT IPL---2540
3096	9EE0	DCLR 034		ERRQ 048	STOP	BR	GO TO 2540 STOP
3098	E71E	DCLR 035	NATPRO	038	NTYPE	BR IF D12=0	CHECK FOR
309A	F71E	DCLR 036		038	NTYPE	BR IF D13=0	1052
309C	83A0	DCLR 037		DYPE 226	PRGCHK	BR	
309E	B16A	DCLR 038	NTYPE	ERRQ 110	GUESS	BAL	RST 2540 CMD.INTLK.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
30A0	9D52	DCLR 039		DPTQ 019	TAKEIN	BR ** TRY FOR INT.	FOR 1403,2540,-----GO TO SFT STP
30A2	2425	DCLR 041	BURST			GO=0\$K20	SET PROGRAM CHECK
30A4	54E9	DCLR 042				HO=GO	MOVE CHANNEL STATUS
30A6	15C5	DCLR 043	ADRXYZ			G1=G1*-KCO	RST CDA & CC FLAGS
30A8	1573	DCLR 044				G1=G1*-K07	RST CODED OP BITS
30AA	CC80	DCLR 045		048	PRGECC	BR IF GS BIT0=0	BR IF CDA FLAG OFF
30AC	2B00	DCLR 046				SET GA K=00	DROP ALL TAGS OUT
30AE	945A	DCLR 047		DCLB 191	BSTCDA	BR	BR TO DCLB TO RTN FROM TRAP
30B0	0D10	DCLR 048	PRGECC			RST GB K=01	RESET SUPR-0
30B2	3413	DCLR 049				GO=GO\$K01	SET STATUS RECEIVED INDICATOR
30B4	2205	DCLR 050				VO=0\$K00	SET 00 UNIT STATUS
30B6	7602	DCLR 051				STH D DA,88	STORE UNIT IDENT HWD IN IB
30B8	2D40	DCLR 052				SET GB K=04	SET INT BUFFER LATCH
30BA	C681	DCLR 053		057	HIPRTP	BR IF MMSK0=1	BR IF CHNL HI PRIORITY TRAP
30BC	E681	DCLR 054		057	HIPRTP	BR IF MMSK2=1	BR IF CHNL LO TRP PRIORITY
30BE	2B00	DCLR 055				SET GA K=00	RST TAGS OUT
30C0	A786	DCLR 056		DCLC 184	ADR22A	BR	BR TO SET CMD-0 & TMOUT ON OPI DW
3080	0240	DCLR 057	HIPRTP			RTN G MMSK0=0	RTN FROM TRAP
0476	3425	DCLR 106	WRAP			GO=GO\$K20	SET PROGRAM CHECK-MEMORY WRAP
0478	9616	DCLR 107		109	WRPPRT	BR	
1614	3415	DCLR 108	PROTCH			GO=GO\$K10	SET PROTECTION CHECK
1616	15C5	DCLR 109	WRPPRT			G1=G1*-KCO	RST CDA & CC FLAGS
1618	1533	DCLR 110				G1=G1*-K03	RST CODED COMMAND BITS
161A	81CA	DCLR 112		DCLD 018	DECCNT	BR	
228E	875A	DCLR 116	IPLICC	DCLL 021	IPLSTP	BR	BR TO IPL STOP WORD
2254	562F	DCLR 117	INFCHK			VO=MMSK	SAVE MMSK REG
2256	221E	DCLR 118				SET MMSK K=71	SET PRIORITY HOLD LATCH
2258	5C3F	DCLR 119				V1=GS	SAVE GS REG
225A	7222	DCLR 121				STH V DA,8C	TEMP STORE MMSK/GS
225C	5D2F	DCLR 125	ADR102			VO=GT	SAVE GT EXTERNAL CDNS
225E	5E3F	DCLR 126				V1=GD	SAVE GD EXTERNAL CDNS
2260	2B10	DCLR 127				SET GA K=01	SET STP TRAP INHIBIT LATCH
2262	24C7	DCLR 128				GO=0	
2264	25E5	DCLR 129				G1=0\$K80	
2266	3573	DCLR 130				G1=G1\$K07	G=CC87
2268	765A	DCLR 131				STH D G-2	STORE UNIT IDENT HWD IN MS 0086
226A	725A	DCLR 132				STH V G-2	STORE GT/GD IN MS0084
226C	5350	DCLR 133				RDB V1 G+0	READ OUT ERROR COUNT INTO V1
226E	231B	DCLR 134				V1=V1+K01	INCREMENT ERROR COUNT
2270	33C5	DCLR 135				V1=V1\$KCO	SET *LGOUT WAS CHNL ICC' BIT
2272	5622	DCLR 137				RDH D DA,8C	READ OUT MMSK/GS
2274	5729	DCLR 141				VO=D1	MOVE GS TO VO REG
2276	725A	DCLR 142				STH V G-2	STORE GS/ERROR COUNT IN MS0082
2278	577F	DCLR 143				D1=BA	MOVE BA TO D1 REG
227A	7650	DCLR 144				STH D G	STORE MMSK/BA IN MS0080
227C	3573	DCLR 145				G1=G1\$K07	BUILD 0087 ADDRESS
227E	5650	DCLR 146				RDH D G	RESTORE UNIT IDENT HWD IN D
2280	E690	DCLR 150		153	ADR100	BR IF MMSK2=0	BR IF NOT IN TRAP
2282	2B00	DCLR 151				SET GA K=00	RST CHNL TAGS
2284	A294	DCLR 152		155	ADR101	BR	
2290	24F6	DCLR 153	ADR100			SET MODE K=3F	
2292	2B80	DCLR 154				SET GA K=08	SET INL SELECT LATCH
2294	2507	DCLR 155	ADR101			G1=0	ZERO OUT G1

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2296	021E	DCLR 156				RST MMSK K=71	
2298	2323	DCLR 157				V1=0\$K02	SET INTERFACE CONTROL CHECK
229A	0D02	DCLR 158				RST GB K=10	RESET PARITY ERROR LATCH
229C	C38F	DCLR 159	IPLICK	116	IPLICC	BR IF BA BIT4=1	BR IF IPL
229E	2B80	DCLR 160				SET GA K=08	SET INL SELECT LATCH
22A0	C9B8	DCLR 161		174	OPIDWN	BR IF GT BIT4=0	BR IF OP-IN IS ALREADY DOWN
22A2	2B84	DCLR 162				SET GA K=28	SET ADR-OUT AND INL SEL LTCHES
22A4	22C7	DCLR 163				V0=0	INITIALIZE TIMEOUT COUNTER
22A6	C9B8	DCLR 164	ADROO1	174	OPIDWN	BR IF GT BIT4=0	BR IF OP-IN DOWN
22A8	221B	DCLR 165				V0=V0+K01	INCREMENT TMOU COUNTER
22AA	F4A6	DCLR 166		164	ADROO1	BR IF AC=0	BR IF NO TIMEOUT
22AC	2D10	DCLR 167				SET GB K=01	SET SUPR-OUT
22AE	3B80	DCLR 168				SET GA K=88	DROP OP-OUT
22B0	221B	DCLR 169	WTOPI2			V0=V0+K01	INCREMENT TIMEOUT COUNTER
22B2	C9B8	DCLR 170		174	OPIDWN	BR IF GT BIT4=0	BR IF OP-IN DOWN
22B4	F4B0	DCLR 171		169	WTOPI2	BR IF AC=0	BR IF NO TIMEOUT
22B6	0D10	DCLR 172				RST GB K=01	RST SUPR-OUT
22B8	C687	DCLR 174	OPIDWN	187	CHAING	BR IF MMSK0=1	BR IF CHNL HI TRP PRIORITY
22BA	E6E7	DCLR 175		187	CHAING	BR IF MMSK2=1	BR IF CHNL LO TRP PRIORITY
22BC	F207	DCLR 179		187	CHAING	BR IF DO BIT7=1	BR IF ENTRY WAS VIA CHAINING
22BE	7222	DCLR 180				STH V DA,8C	TEMP STORE UNIT & CHNL STATUS
22C0	2B00	DCLR 181				SET GA K=00	DROP ALL TAGS OUT
22C2	2400	DCLR 182				SET MODE K=00	SET CPU MODE & ZONE
22C4	5E22	DCLR 183				RDH H DA,8C	READ OUT UNIT/CHNL STATUS
22C5	1000	DCLR 184				RST SO	RST RETURN FLAG IF ON
22C8	B0C8	DCLR 185		010	ADROO3	BR	BR TO STORE CSW (STATUS ONLY)
2286	5349	DCLR 187	CHAING			GO=V1	MOVE CHNL STATUS
2288	3613	DCLR 188				DO=DO\$K01	SET SOURCE OF ENTRY WAS CHAIN
228A	1D00	DCLR 189				RST GB K=80	RESET CDA REQ LTCH
228C	BCA6	DCLR 190		043	ADRXYZ	BR	
03B0	2343	DCLR 196	CCCXXX			V1=0\$K04	SET CHNL CONTROL CHECK
03B2	5612	DCLR 197				RDH D DA,8A	RD OUT UNIT IDENT HWD
03B4	3613	DCLR 198				DO=DO\$K01	SET SOURCE OF ENTRY WAS CHAIN
03B6	A29C	DCLR 199		159	IPLICK	BR	
		DCLR 200	*			*****	*****

***** END OF DCLR (CHANNEL ERROR) *****

 * CROSS REFERENCE FOR CSECT DCLR *

DCLR 002	DCLB 101	DCLB 104			
DCLR 004	DCLR 002				
DCLR 008	DCLA 068				
DCLR 010	DCLR 185				
DCLR 013	DCLR 0C4				
DCLR 031	DCLR 029				
DCLR 035	DCLR 033				
DCLR 038	BWRP 087	BWRP 088	DCLR 035	DCLR 036	
DCLR 041	DCLR 013				
DCLR 043	DCLR 150				
DCLR 048	DCLH 029	DCLR 045			
DCLR 057	DCLR 053	DCLR 054			
DCLR 106	BWRP 019				
DCLR 108	BWRP 028				
DCLR 109	DCLR 1C7				

* CROSS REFERENCE FOR CSECT DCLR *

DCLR 116	DCLR 159				
DCLR 117	DCLC 038	DCLC 112	DCLC 174	DCLE 027	DCLE 045
DCLR 153	DCLR 150				
DCLR 155	DCLR 152				
DCLR 159	DCLR 159				
DCLR 164	DCLR 166				
DCLR 169	DCLR 171				
DCLR 174	DCLR 161	DCLR 164	DCLR 170		
DCLR 187	DCLR 174	DCLR 175	DCLR 179		
DCLR 196	BMCK 037				

DCLT DESCRIPTIVE TEXT

CHANNEL TEST I/O AND I/O INTERRUPT

BURST CHANNEL ENTRY POINTS

TESTIO -- FROM -DCLA- I/O INSTRUCTIONS ROUTINE WHEN A TEST I/O INSTRUCTION IS DECODED.

INTRIX -- FROM -DCLC- TO RESTORE THE PREVIOUS CONTENTS OF THE H-REGISTER BEFORE THE MICROPROGRAM RETURNS TO I-CYCLES.

INTENT -- FROM -BSWI- WHEN AN I/O INTERRUPT IS DETECTED.

BYTE CHANNEL ENTRY POINTS

ENTRY1 -- FROM -DCLA- I/O INSTRUCTIONS ROUTINE WHEN AN ACTIVE TEST I/O IS DECODED.

ENTRY2 -- FROM -DCLA- WHEN AN INACTIVE TEST I/O IS DECODED.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLT 001	T			CHANNEL TEST I/O AND I/O INTERRUPT	H.E.BERKEBILE
2F02	C6C9	DCLT 003	TESTIO	009	ACTIVE	BR IF D0 BIT0=1	BR IF ACTIVE BIT ON
2F04	2007	DCLT 004				UO=0	SET UO FOR TEST I/O
2F06	A344	DCLT 005		DCLH 006	ADRO	BR	GO SET UP MODE AND ZONE
		DCLT 006	*				RETRIEVE STATUS FROM
		DCLT 007	*				UNIT
2F00	AC74	DCLT 008	ADRO	CCOM 161	CC2B	BR	BR TO CC2 SET
2F08	24F6	DCLT 009	ACTIVE			SET MODE K=3F	SET CHANNEL MODE-ZONE
2F0A	EC80	DCLT 010		008	ADRO	BR IF GS2=0 ** BR IF FILE IS ACTIVE-NOT CHANNEL	
2F0C	F000	DCLT 011		008	ADRO	BR IF G0 BIT7=0	BR IF NO STATUS RECEIVED
2F0E	2400	DCLT 012				SET MODE K=00	SWITCH BACK TO CPU MODE & ZONE
2F10	5202	DCLT 013				RDH V DA, 88	READ INT BUFFER
2F12	6311	DCLT 014				V1=V10U1	ADDRESS MATCH
2F14	C480	DCLT 015		008	ADRO	BR IF ZNZ	BR IF NO MATCH
2F16	24F6	DCLT 016				SET MODE K=3F	SET BACK TO CHNL MODE & ZONE
2F18	C11C	DCLT 017	ADRI	019	SKIP	BR IF G14=0	BR IF NO PCI
2F1A	3485	DCLT 018				GO=GO\$K80	SET PCI
2F1C	70F2	DCLT 019	SKIP			STH U DA, BE	SAVE COUNT
2F1E	5419	DCLT 020				U1=GO	MOVE STATUS
2F20	5209	DCLT 021				UO=VO	UNIT STATUS
2F22	4452	DCLT 022				RDH G DC, 9A	READ ZERO CONSTANT INTO G
2F24	7432	DCLT 023				STH G CA, 8E	ZERO OUT UNIT ADDRESS
2F26	0DC4	DCLT 024				RST GB K=20	RST BURST LATCH
2F28	70E2	DCLT 025	REDUC			STH U DA, BC	STORE STATUS
2F2A	0D40	DCLT 026				RST GB K=04	RST INT
2F2C	C21E	DCLT 027	SAVE1			RST MMSK K=71	ALLOW TRAPS
2F2E	2400	DCLT 028				SET MODE K=00	
2F30	4652	DCLT 029				RDH D DC, 9A	READ ZERO CONSTANT INTO D
2F32	2B87	DCLT 031				T1=0\$K88	GET
2F34	52AC	DCLT 032				ISK STPO AS, T+2	PROTECT
2F36	526F	DCLT 033				DO=STPO	KEY
2F38	566B	DCLT 034				DO=DOH	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2F3A	5212	DCLT 036				RDH V DA,8A	CCW ADDRESS
2F3C	50F2	DCLT 037				RDH U DA,BE	CDUNT
2F3E	5EE2	DCLT 038				RDH H DA,BC	STATUS
2F40	1F13	DCLT 039				H1=H1*-K01	RST INT. BIT
2F42	A024	DCLT 040		CCOM 127	CSWADR	BAL	STORE CSW SO=1FOR INT.RTN
2F44	B354	DCLT 041	INTRTX	CCOM 186	RESTRH	BAL	RESTORE CPU H1
2F46	5A22	DCLT 042				RDH T DA,8C	UNIT ADDRESS FOR INT.
2F48	2A13	DCLT 043				TO=0\$K01	FIX CHANNEL ADDRESS
2F4A	2400	DCLT 044	INTRTN			SET MODE K=00	
2F4C	A43E	DCLT 045		DGHN 075	USECOM	BR	GO TAKE INTERRUPT
		DCLT 046	*			*****	
22FA	3000	DCLT 047	INTENT			SET SO	SET THE RETURN FLAG
22FC	2480	DCLT 048				SET MODE K=08	FILE MODE
22FE	5432	DCLT 049				RDH G DA,8E	
2300	C407	DCLT 050		054	ACTON	BR IF GO BIT0=1	
2302	E98A	DCLT 051		056	INACHL	BR IF FFI6=0	
2304	A5FC	DCLT 052	FINT	FINT 002	INTRPT	BR	
		DCLT 053	*			*****	
2306	D985	DCLT 054	ACTON	052	FINT	BR IF FFI5=1	
2308	C985	DCLT 055		052	FINT	BR IF FFI4=1	BR IF PCI LATCH IS ON
230A	5A02	DCLT 056	INACHL			RDH T DA,88	RD UNIT IDENT HWD FROM IB
230C	2A15	DCLT 057				TO=0\$K10	SET TO TO TEST I/O INDICATOR
230E	7A22	DCLT 058				STH T DA,8C	TEMP STORE UNIT ADDRESS
2310	2A13	DCLT 059				TO=0\$K01	SET TO TO CHNL ADDRESS
2312	C446	DCLT 060		DCLH 007	ADROA	BR IF GO BIT0=0	BR IF NOT ACTIVE
2314	2A07	DCLT 061				TO=0	
2316	2413	DCLT 062				GO=0\$K01	SET GO FOR INTERRUPT
2318	24F6	DCLT 063				SET MODE K=3F	SET CHNL MODE & ZONE
231A	221E	DCLT 064				SET MMSK K=71	SET PRIORITY HOLD
231C	F056	DCLT 065		068	ACTPCI	BR IF GO BIT7=0	BR IF NOT FROM INT
231E	1413	DCLT 066				GO=GO*-K01	RESET GO BIT7
2320	AF18	DCLT 067		017	ADR1	BR	BR TO CSW STORE
2356	1583	DCLT 068	ACTPCI			G1=G1*-K08	TURN OFF PCI IN OPFLG
2358	FCDC	DCLT 069		071	NOCDZ	BR IF GS3=0	BR IF NO PARITY ERROR
235A	3483	DCLT 070				GO=GO\$K08	SET DATA CHECK
235C	0D40	DCLT 071	NOCDZ			RST GB K=04	RST INT. LATCH
235E	2470	DCLT 072				SET MODE K=07	SET CHANNEL ZONE-CPU MODE
2360	74A2	DCLT 073				STH G DA,AC	STORE STATUS
2362	70F2	DCLT 074				STH U DA,BE	STORE COUNT
2364	2400	DCLT 075				SET MODE K=00	
2366	5BA2	DCLT 076				RDB T1 DA,AC	GET CHAN STATUS
2368	3B85	DCLT 077				T1=T1\$K80	SET PCI
236A	7AE2	DCLT 078				STH T DA,BC	STORE STATUS WITH PCI
236C	AF2C	DCLT 079		027	SAVE1	BR	

 * CROSS REFERENCE FOR CSECT DCLT *

DCLT 003 DCLA 110
 DCLT 008 DCLT 010 DCLT 011 DCLT 015
 DCLT 009 DCLT 003
 DCLT 017 DCLT 067
 DCLT 019 DCLT 017
 DCLT 027 DCLT 079

* CROSS REFERENCE FOR CSECT DCLT *

DCLT 041	DCLC 058	
DCLT 047	BSWI 093	
DCLT 052	DCLT 054	DCLT 055
DCLT 054	DCLT 050	
DCLT 056	DCLT 051	
DCLT 068	DCLT 065	
DCLT 071	DCLT 069	

DCOM -- COMMON I/O ROUTINES

THESE COMMON ROUTINES ARE EMPLOYED BY THE I/O ROUTINES FOR THE NATIVE DEVICES -- PRINTER, READER-PUNCH, AND PR-KB.

ENTRIES

PCIBR -- USED TO TRY FOR A PROGRAM-CONTROLLED INTERRUPT. IF THE PCI FLAG IS OFF, THE MICROPROGRAM RETURNS TO THE LINK ADDRESS. IF THE PCI FLAG IS ON, THE PCI-ALONE INDICATOR BIT IS SET FOR THE INTERRUPT BUFFER. FROM THIS POINT, PCI IS HANDLED THE SAME AS THE NORMAL INTERRUPT.

INTRPT -- NORMAL INTERRUPT ENTRY, I.E., WHEN CHANNEL-END AND/OR DEVICE-END STATUS IS READY. IF THE INTERRUPT IN I.B. BIT IS ON, THE DEVICE ALREADY HAS CONTROL OF THE I.B. OTHERWISE, THE ROUTINE MUST TEST TO SEE IF THE I.B. IS AVAILABLE. IF THE I.B. IS NOT AVAILABLE, THE INTERRUPT IS STACKED AND ANOTHER REQUEST WILL OCCUR WHEN THE I.B. IS AVAILABLE.

IF THE I.B. IS AVAILABLE, AN INTERRUPT IS INITIATED AS FOLLOWS.

1. SET CHANNEL 0 INTERRUPT LATCH, S7.
2. SET I.B. BIT IN CHANNEL-STATUS REG.
3. STORE UCW ADDRESS IN INTERRUPT BUFFER.
4. STORE PCI-ALONE BIT IN I.B. IF THIS BIT WAS TURNED ON AT -PCIBR- ENTRY.

AFTER THE INTERRUPT IS INITIATED, THE MICROPROGRAM RETURNS TO THE LINK ADDRESS.

CHANOK
REDSTR -- USED TO UPDATE THE UCW. CHANNEL STATUS, FLAGS/OP, UNIT STATUS, AND COUNT ARE STORED IN THE UCW. IF COMMAND CHAINING, AND DEVICE-END IS SET, ENTRY IS TO -CHANOK- TO CLEAR THE UNIT STATUS REGISTER.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCCM 001	T			COMMON I/O ROUTINES	
		DCCM 002	*				
		DCOM 003	*			THIS ROUTINE TRIES TO PUT THE NATIVE DEVICE INTO THE INTERRUPT	
		DCOM 004	*			BUFFER FOR PCI OR STATUS.	
		DCOM 005	*				
3106	C140	DCOM 006	PCIBR	035	RETURN	BR IF G14=0	BR IF NO PCI
3108	2683	DCOM 007				DO=0\$K08	SET PCI ALONE
310A	F017	DCOM 008	INTRPT	014	ITSME	BR IF G07=1	BR IF SAME IN INT BUFFER
310C	221E	DCOM 009				SET MMSK K=71	BLOCK TRAPS
310E	C7B5	DCOM 010		029	IBTAKN	BR IF BA BIT0=1	BR IF IB FULL
3110	2010	DCOM 011				SET S7	TAKE IT
3112	021E	DCOM 012				RST MMSK K=71	ALLOW TRAPS
3114	3413	DCOM 013				GO=GO\$K01	SET IB BIT
3116	76D2	DCOM 014	ITSME			STH D DA,BA	STORE ADDRESS IN IB
3118	1683	DCOM 015				DO=DO*-K08	
311A	F727	DCOM 016	RSTREQ	020	RDTWI	BR IF D1 BIT3=1	RST REQUEST, BR IF RDR OR TW
311C	E723	DCOM 017		023	PCH1	BR IF D1 BIT2=1	BR IF PCH
311E	0D08	DCOM 018				RST PRA K=40	RST PTR REQUEST
3120	128E	DCOM 019				RTN	
3126	E72D	DCOM 020	RDTWI	025	TWI	BR IF D1 BIT2=1	BR IF TW
3128	0B20	DCOM 021				RST R K=02	RST RDR REQUEST
312A	128E	DCOM 022				RTN	
3122	0F20	DCOM 023	PCH1			RST P K=02	RST PCH REQUEST
3124	128E	DCOM 024				RTN	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
312C	0F24	DCOM 025	TW1			RST TA K=22	RST SHARE MICRO FORCE
312E	C832	DCOM 026		028	TYREXT	BR IF P04=0	
3130	1C83	DCOM 027				PO=P0*-K08	RST END STAT
3132	128E	DCOM 028	TYREXT			RTN	
3134	021E	DCOM 029	IBTAKN			RST MMSK K=71	SOMEONE ELSE IN IB RST MMSK
3136	1683	DCOM 030				DO=D0*-K08	RST PCI ALONE
3138	D541	DCOM 031		035	RETURN	BR IF G11=1	DONT QUE PCI CMND CHAIN
313A	F747	DCOM 032		036	RDTW2	BR IF D1 BIT3=1	SET QUE AND REQ., R IF RDR OR TW
313C	E743	DCOM 033		039	PCH2	BR IF D1 BIT2=1	BR IF PCH
313E	2D88	DCOM 034				SET PRA K=48	SET PTR REQUEST AND QUE
3140	128E	DCOM 035	RETURN			RTN	
3146	E701	DCOM 036	RDTW2	041	TW2	BR IF D1 BIT2=1	BR IF TW
3148	2B30	DCOM 037				SET R K=03	SET RDR REQUEST AND QUE
314A	128E	DCOM 038				RTN	
3142	2F30	DCOM 039	PCH2			SET P K=03	SET PCH REQUEST AND QUE
3144	128E	DCOM 040				RTN	
3100	C804	DCOM 041	TW2	043	TWREXT	BR IF P04=0	
3102	2F04	DCOM 042				SET TA K=20	SET 1052 MICRO FORCE
3104	128E	DCOM 043	TWREXT			RTN	
		DCOM 044	*				
		DCOM 045	*				
		DCOM 046	*				
		DCOM 047	*				
						THIS ROUTINE STORES THE CHNL STATUS, FLAGS, UNIT STATUS AND THE COUNT OF THE NATIVE DEVICE INTO ITS UCW.	
2566	2E07	DCOM 048	CHANOK			H0=0	
2568	7468	DCOM 049	REDSTR			STH G AS,D+2	
256A	7E68	DCOM 050				STH H AS,D+2	
256C	7060	DCOM 051				STH U AS,D+0	STORE COUNT
256E	C573	DCOM 052		054	RETUN	BR IF G1 BIT0=1	BR IF CDA
2570	2AC7	DCOM 053				TO=0	RST TO
2572	128E	DCOM 054	RETUN			RTN	

 * CROSS REFERENCE FOR CSECT DCOM *

DCOM 006	DPTR 067	DYPE 040	
DCOM 008	DPTQ 019	DYPE 012	DYPE 085
DCOM 014	DCOM 008		
DCOM 016	ERRQ 036	ERRQ 050	
DCOM 020	DCOM 016		
DCOM 023	DCOM 017		
DCOM 025	DCOM 020		
DCOM 028	DCOM 026		
DCOM 029	DCOM 010		
DCOM 035	DCOM 006	DCOM 031	
DCOM 036	DCOM 032		
DCOM 039	DCOM 033		
DCOM 041	DCOM 036		
DCOM 043	DCOM 041		
DCOM 048	DPTQ 036	DYPE 153	
DCOM 049	DPTQ 020	DPTS 072	DYPE 227 ERRQ 046
DCOM 054	DCOM 052		

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DMCS 001	T			MCS LOAD ROUTINE KREPPLE	
		DMCS 002	*				
2F84	52A2	DMCS 003	NOTLEN			RDH V DA,AC	RESTORE DATA ADDRESS
2F86	6DD3	DMCS 004				P1=P1+P1	UPDATE PCCL CONSTANT
2F88	DD16	DMCS 005		022	TRYLEN	BR IF P1 BIT1=0	BR IF CONSTANT NOT LENGTH 240
2F8A	2FF5	DMCS 006				H1=0\$KFO	SET CHAIN LENGTH TO 240
2F8C	AFB0	DMCS 007		035	SETHDW	BR	GO SET PCCL FOR LENGTH 240
		DMCS 008	*				
2F78	2E85	DMCS 009	MCSNOF			H0=0\$K80	SET NO FOLDING SWITCH
2F7A	74E2	DMCS 010	MCSRTN			STH G DA,BC	SAVE G REG
2F7C	7EF2	DMCS 011				STH H DA,BE	SAVE H REG
2F7E	2F10	DMCS 012				SET PRB K=01	SET DIAGNOSTIC MODE 4
2F80	EBCF	DMCS 013		018	LOADOK	BR IF H1 BIT6=1	BR IF GATE LOAD WAS ON
2F82	8612	DMCS 014		DPTC 024	CMDREJ	BR	GATE LOAD WAS NOT ON REJ MCS CMD
		DMCS 015	*				
		DMCS 016	*				DETERMINE CHAIN LENGTH ROUTINE
		DMCS 017	*				
2F8E	2D13	DMCS 018	LOADOK			P1=0\$K01	BUILD INITIAL PCCL CONSTANT
2F90	72A2	DMCS 019				STH V DA,AC	SAVE INITIAL DATA ADDRESS
2F92	2427	DMCS 020				G0=0\$K22	BUILD ADDRESS OF THE
2F94	25A3	DMCS 021				G1=0\$K0A	TABLE OF CHAIN LENGTHS
2F96	4F48	DMCS 022	TRYLEN			RDB H1 CS,G+1	READ CHAIN LENGTH FROM TABLE
2F98	5FC9	DMCS 023				P0=H1	SET COUNTER TO CHAIN LENGTH
2F9A	4826	DMCS 024				I=V	BUILD ADDRESS
2F9C	69FB	DMCS 025				I1C=I1+H1	EQUAL TO
2F9E	F5A2	DMCS 026		028	COMPAR	BR IF S3=0	DATA ADDRESS
2FA0	281B	DMCS 027				I0=I0+K01	PLUS CHAIN LENGTH
2FA2	5B38	DMCS 028	COMPAR			RDB T1 V+1	READ DATA BYTE 1
2FA4	5198	DMCS 029				RDB U1 I+1	READ DATA BYTE 2
2FA6	2C1B	DMCS 030				P0=P0+K01	INCREMENT COUNTER
2FA8	6B11	DMCS 031				T1=T1=U1	COMPARE DATA BYTES
2FAA	C484	DMCS 032		003	NOTLEN	BR IF ZNZ	BRANCH IF DATA BYTES NOT EQUAL
2FAC	0CFD	DMCS 033				Z=P0=KFO	TEST FOR LAST COMPARE
2FAE	C4A2	DMCS 034		028	COMPAR	BR IF ZNZ	GO BACK IF NOT LAST COMPARE
2FB0	49DF	DMCS 035	SETHDW			PCCL=P1	SET HDWR LATCH TO CHAIN LENGTH
2FB2	27FF	DMCS 036				D1=D1+KFF	SET D TO PCCL SAVE ADDRESS OF C7
2FB4	7D60	DMCS 037				STB P1 AS,D	STORE HDWR BYTE FOR SYSTEM RESET
2FB6	52A2	DMCS 038				RDH V DA,AC	RESTORE INITIAL DATA ADDRESS
		DMCS 039	*				
		DMCS 040	*				CLEAR MCS TABLE ROUTINE
		DMCS 041	*				
2FB8	2C75	DMCS 043				P0=0\$K70	BUILD MCS XLATE TABLE ADDRESS
2FBA	27F7	DMCS 047				D1=0\$KFF	BUILD CLEAR CHARACTER OF FF
2FBC	77C8	DMCS 048	CLEAR			STB D1 AS,P+1	CLEAR TRANSLATE
2FBE	E83C	DMCS 049		048	CLEAR	BR IF P0 BIT6=0	TABLE TO FF
2FC0	2105	DMCS 050				U1=0\$K00	BUILD 00 CHARACTER
2FC2	71C0	DMCS 051				STB U1 AS,P	STORE 00 IN POS 00 XLATE TABLE
2FC4	B2C6	DMCS 052		061	STZERO	BR	GO STORE 00 IN POS 40 XLATE TBL
		DMCS 053	*				
		DMCS 054	*				LOAD MCS TABLE ROUTINE
		DMCS 055	*				
32BC	5D38	DMCS 056	RDLOOP			RDB P1 V+1	READ CHAIN IMAGE BYTE
32BE	3E43	DMCS 057				H0=H0\$K04	TURN ON NEW CHARACTER SWITCH

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
32C0	211B	DMCS 058				U1=U1+K01	INCREMENT PRINT POSITION COUNTER
32C2	71C0	DMCS 059	STLOOP			STB U1 AS,P	STORE PRINT POS NUMBER IN TABLE
32C4	CE4D	DMCS 060		064	NOFOLD	BR IF H0 BIT0=1	BR AROUND FOLD ROUTINE IF ON
32C6	2D4D	DMCS 061	STZERO			P1=P1+K40	PLUS 40 HEX TO ADDRESS FOR FOLD
32C8	264D	DMCS 062				DO=DO+K40	ADD 40 TO NUMBER OF FOLDS CTR
32CA	F4C2	DMCS 063		059	STLOOP	BR IF DYN BIT3=0	STAY IN LOOP 4 TIMES
32CC	CBE7	DMCS 064	NOFOLD	081	CKLAST	BR IF PRD BIT4=1	BRANCH IF NOT DUALING
32CE	CF67	DMCS 065		081	CKLAST	BR IF H1 BIT0=1	BR AROUND DUAL RTN IF 240 CHAR
32D0	DA66	DMCS 066		081	CKLAST	BR IF H0 BIT5=0	BR IF NEW CHARACTER SWITCH OFF
		DMCS 067	*				
		DMCS 068	*				DUALING ROUTINE
		DMCS 069	*				
32D2	1E43	DMCS 070				H0=H0*-K04	RESET NEW CHARACTER SWITCH
32D4	2515	DMCS 071				G1=0\$K10	SET REG TO DUALED PAIR LIST
32D6	C167	DMCS 072	RDLIST	081	CKLAST	BR IF G1 BIT4=1	BR WHEN REG POINTS PAST LIST
32D8	4948	DMCS 073				RDB I1 CS,G+1	GET ONE OF THE DUALED CHARACTERS
32DA	69D1	DMCS 074				I1=I1PI	COMPARE TO CHAIN IMAGE CHARACTER
32DC	C4D6	DMCS 075		072	RDLIST	BR IF ZNZ	BR IF NOT EQUAL
32DE	F163	DMCS 076		078	ADDROK	BR IF G1 BIT7=1	BR IF MATCH WAS FIRST OF PAIR
32E0	6446	DMCS 077				G=G-2	SET ADDR TO CHAR TO BE DUALED
32E2	4D40	DMCS 078	ADDROK			RDB P1 CS,G	GET TABLE ADDR OF DUALED CHAR
32E4	B2C2	DMCS 079		059	STLOOP	BR	GO STORE PRINT POSITION NUMBER
		DMCS 080	*				IN DUAL CHARACTER POSITION
32E6	51B9	DMCS 081	CKLAST			T1=U1	TEST FOR
32E8	6BF1	DMCS 082				T1=T1HI	LAST CHARACTER
32EA	C4BC	DMCS 083		056	RDLOOP	BR IF ZNZ	BRANCH IF NOT LAST CHARACTER
		DMCS 084	*				
		DMCS 085	*				END HOUSEKEEPING
		DMCS 086	*				
32EC	54E2	DMCS 087				RDH G DA,BC	RESTORE G REG
32EE	5EF2	DMCS 088				RDH H DA,BE	RESTORE H REG
32F0	4642	DMCS 089				RDH D DC,98	D=COC8---1403 UCW ADDRESS
32F2	2105	DMCS 090				U1=0\$K00	ZERO COUNT
32F4	2E43	DMCS 091				H0=0\$K04	SET DEVICE END
32F6	8D2E	DMCS 092		DPTR 054	CHEND	BR	GO TO CHANNEL END ROUTINE
		DMCS 093	*				
		DMCS 094	ATABLE	ADDR=220A			
220A	1028	DMCS 095	C			XCTL' 1028303C50784C5D6C4D7C7D7B7E'	
		DMCS 096	AEND				

 * CROSS REFERENCE FOR CSECT DMCS *

DMCS 003	DMCS 032	
DMCS 009	DPTC 060	
DMCS 010	DPIC 047	
DMCS 018	DMCS 013	
DMCS 022	DMCS 005	
DMCS 028	DMCS 026	DMCS 034
DMCS 035	DMCS 007	
DMCS 048	DMCS 049	
DMCS 056	DMCS 083	
DMCS 059	DMCS 063	DMCS 079
DMCS 061	DMCS 052	

* CROSS REFERENCE FOR CSECT DMCS *

DMCS 064	DMCS 060
DMCS 072	DMCS 075
DMCS 078	DMCS 076
DMCS 081	DMCS 064 DMCS 065 DMCS 066 DMCS 072

DPTC DESCRIPTIVE TEXT

ENTRY POINTS

NOOP - ENTRY IS MADE TO THIS LABEL TO SET CHANNEL END AND DEVICE END FOR ANY OF THE NO-OP COMMANDS ISSUED FOR THE PRINTER OR READER-PUNCH.

CMDREJ - WHEN A LOAD MCS AND FOLD, OR A LOAD MCS AND NO FOLD COMMAND, IS NOT PRECEDED BY A GATE LOAD COMMAND, ENTRY IS MADE TO THIS LABEL TO REJECT THE COMMAND.

COMHI X - ENTRY IS MADE TO THIS BRANCH SET FROM THE -DPTS- ROUTINE TO DECODE THE PRINTER COMMAND.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DPTC 001	T			1403 COMMAND DECODE AND CHECK READ	
		DPTC 002	*				
		DPTC 003	*				
		DPTC 004	*				
		DPTC 006	*			GATE LOAD COMMAND	
060C	F213	DPTC 007	GATELD	024	CMDREJ	BR IF DO BIT7=1**BR IF LAST COMMAND CHAIND	
060E	3F23	DPTC 008				H1=H1\$K02 **SET GATE LOAD TAG	
0610	861C	DPTC 009		021	NOOP	BR **BR TO NO-OP	
		DPTC 010	*			ALLOW DATA CHECK COMMAND	
0604	EB09	DPTC 011	ADATCK	013	FIRST	BR IF H16=1	BR IF GATE LOAD IS ON
0606	F213	DPTC 012		024	CMDREJ	BR IF D07=1	BR IF NOT FIRST IN CHAIN
0608	1F43	DPTC 013	FIRST			H1=H1*-K04	RST BLOCK DATA CHK
060A	861C	DPTC 014		021	NOOP	BR	GO UPDATE SENSE BYTE
		DPTC 015	*			BLOCK DATA CHECK COMMAND	
0616	EB18	DPTC 016	BDATCK	018	ISFIRS	BR IF H16=1	BR IF GATE LOAD IS ON
0618	F213	DPTC 017		024	CMDREJ	BR IF D07=1	BR IF NOT FIRST IN CHAIN
061A	3F43	DPTC 018	ISFIRS			H1=H1\$K04	SET BLOCK DATA CHECK
		DPTC 020	*			NO-OP COMMAND	
061C	2EC3	DPTC 021	NOOP			H0=0\$K0C	SET CH END, DEV END
061E	8D58	DPTC 022		DPTR 080	IMMEDI	BR	
0612	3F85	DPTC 024	CMDREJ			H1=H1\$K80	SET COMMAND REJ
0614	9D3A	DPTC 025		DPTS 064	ISLCHK	BR	**BR TO INITIAL SELCTN CHK
		DPTC 026	*			*****	
		DPTC 027	*			***1403 COMMAND DECODE ***	
		DPTC 028	*			*****	
1A00	C82E	DPTC 029	COMHI 0	065	COMLOS	BR IF P04=0	
1A02	9835	DPTC 030	COMHI 1	083	COMLOB N	N=PO BITS567	
1A04	8612	DPTC 031	COMHI 2	024	CMDREJ	BR	INVALID
1A06	8612	DPTC 032	COMHI 3	024	CMDREJ	BR	COMMAND
1A08	8612	DPTC 033	COMHI 4	024	CMDREJ	BR	GO TO
1A0A	8612	DPTC 034	COMHI 5	024	CMDREJ	BR	COMMAND
1A0C	8612	DPTC 035	COMHI 6	024	CMDREJ	BR	REJECT
1A0E	9042	DPTC 036	COMHI 7	067	DCCM1	BR	
1A10	C80C	DPTC 037	COMHI 8	035	COMHI 6	BR IF P04=0	SEPERATE LOW INVALID

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1A12	9835	DPTC 038	COMHI 9	083	COMLOB N	N=PO BITS567	
1A14	9835	DPTC 039	COMHI A	083	COMLOB N	N=PO BITS567	
1A16	9835	DPTC 040	COMHI B	083	COMLOB N	N=PO BITS567	
1A18	9835	DPTC 041	COMHI C	083	COMLOB N	N=PO BITS567	
1A1A	9835	DPTC 042	COMHI D	083	COMLOB N	N=PO BITS567	
1A1C	C81A	DPTC 043	COMHI E	042	COMHI D	BR IF P04=0	
1A1E	0C3B	DPTC 044	COMHI F			Z=P00K03	LOOK
1A20	FOA4	DPTC 045		051	BCHECK	BR IF LZNZ	FOR UCB F3,FB
1A22	AF7A	DPTC 047		DMCS 010	MCSRTN	BR	BR TO LOAD UCB FOLD
1A24	0C8B	DPTC 051	BCHECK			Z=P00K0B	
1A26	FC8C	DPTC 052		035	COMHI 6	BR IF LZNZ	BR IF NOT F3 OR FB
1A28	FC2D	DPTC 053		060	LDUCB	BR IF P03=1	COMM IS XB
1A2A	86CC	DPTC 055		007	GATELD	BR	COMM IS LOAD *EB*
1A2C	AF78	DPTC 060	LDUCB	DMCS 009	MCSNOF	BR	**FB** GO LOAD NO FOLD
		DPTC 064	*		COMMAND DECODE	01 THRU OF	
1A2E	9855	DPTC 065	COMLOS	091	LOBRS N	N=PO BITS567	
		DPTC 066	*		COMMAND DECODE	,73 AND 7B	
1042	CC3B	DPTC 067	DCQM1			Z=P00K03	
1044	FOCA	DPTC 068		074	CHECAL	BR IF LZNZ	
1046	8616	DPTC 070		016	BDATCK	BR	BLOCK DATA CHECK COMM
104A	0C8B	DPTC 074	CHECAL			Z=P00K0B	
104C	FOC8	DPTC 075		081	BADCOM	BR IF LZNZ	
104E	86C4	DPTC 077		011	ADATCK	BR	ALLOW DATA CHECK
1048	8612	DPTC 081	BADCOM	024	CMDREJ	BR	REJECT
		DPTC 082	*		COMMAND DECODE		
1AA0	8612	DPTC 083	COMLOB 0	024	CMDREJ	BR	INVALID
1AA2	8CDC	DPTC 084	COMLOB 1	DPTR 006	WRITE	BR	WRITE WITH VALID MOD
1AA4	8612	DPTC 085	COMLOB 2	024	CMDREJ	BR	INVALID
1AA6	8054	DPTC 086	COMLOB 3	DPTR 077	CARCTL	BR	CARRIAGE CTRL
1AA8	8612	DPTC 087	COMLOB 4	024	CMDREJ	BR	INVALID
1AAA	8612	DPTC 088	COMLOB 5	024	CMDREJ	BR	INVALID
1AAC	8612	DPTC 089	COMLOB 6	024	CMDREJ	BR	INVALID
1AAE	8612	DPTC 090	COMLOB 7	024	CMDREJ	BR	INVALID
1AC0	8612	DPTC 091	LOBRS 0	024	CMDREJ	BR	INVALID
1AC2	8CDC	DPTC 092	LOBRS 1	DPTR 006	WRITE	BR	WRITE WITH VALID MOD
1AC4	8612	DPTC 093	LOBRS 2	024	CMDREJ	BR	INVALID
1AC6	861C	DPTC 094	LOBRS 3	021	NOOP	BR	NO OP
1AC8	8F66	DPTC 095	LOBRS 4	DPTS 083	SENSE1	BR	SENSE
1ACA	8612	DPTC 096	LOBRS 5	024	CMDREJ	BR	INVALID
1ACC	8612	DPTC 097	LOBRS 6	024	CMDREJ	BR	INVALID
1ACE	8612	DPTC 098	LOBRS 7	024	CMDREJ	BR	INVALID
		DPTC 099	*				*****
		DPTC 100	*				*****
		DPTC 101	*				*** PRINT CHARACTER CTRL LENGTH ***
		DPTC 102	ATABLE	ADDR=00C6			
00C6	0004	DPTC 103	HAUX				
		DPTC 104	AEND				
		DPTC 105	*	1403	TRANSLATE TABLE IN AUX STORAGE	*****	
		DPTC 107	ATABLE	ADDR=7000			
7000	0025	DPTC 111	C				XAUX*0025262728292A2B2C2D2E2F30182E0016191A1B1C1D*
7016	1E1F	DPTC 112	C				XAUX*1E1F20212223243000C0220D0E0F1011121314151617*
702C	1800	DPTC 113	C				XAUX*180017000A0102030405060708090A0B0C0C0B000025*
7042	2627	DPTC 114	C				XAUX*262728292A2B2C2D2E2F30182E0016191A1B1C1D1E1F*

DPTQ DESCRIPTIVE TEXT

ENTRY POINTS

PTRREQ - ENTRY IS MADE AT THIS LABEL FROM THE -BSWI- ROUTINE. BIT 7 OF THE EXTERNAL (PRS) HAS BEEN TESTED AND FOUND TO BE ON. THIS INDICATES THAT A PRINT REQUEST HAS BEEN GENERATED.

STINST - ENTRY HERE IS TO STORE CHANNEL STATUS, FLAGS, UNIT STATUS, AND COUNT INTO THE UCW.

CHAIOK - WHEN COMMAND CHAINING IS IN EFFECT, THIS ENTRY IS USED TO ZERO UNIT STATUS, STORE UPDATED UCW.

TAKEIN - ENTRY IS MADE HERE TO ATTEMPT TO STORE THE UCW ADDRESS OF THE NATIVE DEVICE INTO THE INTERRUPT BUFFER.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DPTQ 001	T			1403 INTERRUPT REQUEST ROUTINE	
		DPTQ 003	*			*****	
		DPTQ 004	*			*****	
		DPTQ 005	*			INTERRUPT REQUEST ROUTINE MODE IS 18 FROM TYPE	
1D6A	4642	DPTQ 006	PTRREQ			RDH D DC,98	D=COC8---1403 UCW ADDRESS
1D6C	A072	DPTQ 007		ERRQ 080	RDCONT	BAL ** GO GET CHAN.,	UNIT STATUS IN G,H. CT IN U
1D6E	D149	DPTQ 008		023	SECCHK	BR IF G15=1	BR IF ACTIVE
1D70	C44F	DPTQ 009		016	DEVEN1	BR IF G00=1	BR IF SECONDARY ON
1D72	CAF7	DPTQ 010		012	CLEAR	BR IF PRS4=1	BR IF INIT READY ON
1D74	3423	DPTQ 011	IFCHEK			GO=GO\$K02	SET INTERFACE CHECK
1D76	1F45	DPTQ 012	CLEAR			H1=H1*-K40	**RESET INTERVENTION REQD
1D78	3485	DPTQ 013				GO=GO\$K80	SET SECONDARY ON
1D7A	2E43	DPTQ 014				H0=0\$K04	SET DEVICE END
1D7C	9D52	DPTQ 015		019	TAKEIN	BR	
1D4E	CE99	DPTQ 016	DEVEN1	026	CH12	BR IF PRS0=1	BR IF EXT DEV END
1D50	DA74	DPTQ 017		011	IFCHEK	BR IF H05=0	BR IF NO DE STAT
1D52	B10A	DPTQ 019	TAKEIN	DCOM 008	INTRPT	BAL ** GO TRY FOR NORMAL INTERRUPT--NOT PCI	
1D54	A568	DPTQ 020	STINST	DCOM 049	REDSTR	BAL ** GO UPDATE UCW G,H-CHAN. STAT, FLAGS UNIT STAT	
1D56	AAA2	DPTQ 021		BSWI 066	RESTRH	BR	RST MODE--GO BACK TO NAT REQ
1D48	C474	DPTQ 023	SECCHK	011	IFCHEK	BR IF G00=0	BR IF NO SECONDARY
1D4A	CE99	DPTQ 024		026	CH12	BR IF PRS0=1	BR IF EXT DE
1D4C	9D60	DPTQ 025		030	CHAINC	BR	
1D58	B14C	DPTQ 026	CH12	DPTT 071	SENCHK	BAL	GO CHECK SENSE
1D5A	0EC3	DPTQ 027				Z=H0*-K0C	LOOK FOR UNIT CHECK
1D5C	FOE1	DPTQ 028		030	CHAINC	BR IF LZ=0	OR UNIT EXCEPTION
1D5E	1545	DPTQ 029				G1=G1*-K40	RST CHAINING IF ERROR
1D60	D565	DPTQ 030	CHAINC	032	TESTDV	BR IF G11=1	BR IF CHAIN
1D62	9D52	DPTQ 031		019	TAKEIN	BR	
1D64	DA45	DPTQ 032	TESTDV	036	CHAIOK	BR IF H05=1	BR IF DE
1D66	2E07	DPTQ 033				H0=0	CLEAR STATUS
1D68	9D54	DPTQ 034		020	STINST	BR ** STORE UCW - GO BACK TO INRU	
1D44	A566	DPTQ 036	CHAIOK	DCOM 048	CHANDK	BAL ** GO UPDATE UCW CHAN. STAT, FLAGS UNIT STAT	

ADDR	WORD	SEQUENCE NU.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1D46	93CA	DPTQ 037		DCLB 039	CHAIN	BR	***** * CROSS REFERENCE FOR CSECT DPTQ * *****
DPTQ 006	BSWI	065					
DPTQ 011	DPTQ	017	DPTQ 023				
DPTQ 012	DPTQ	010					
DPTQ 016	DPTQ	0C9					
DPTQ 019	BWRP	086	DCLR 039	DPTQ 015	DPTQ 031	DPTR 072	DPTS 069 ERRQ 063
DPTQ 020	DPTQ	034	DPTR 073	ERDR 073	EXFR 142		
DPTQ 023	DPTQ	008					
DPTQ 026	DPTQ	016	DPTQ 024				
DPTQ 030	DPTQ	025	DPTQ 028				
DPTQ 032	DPTQ	030					
DPTQ 036	DPTQ	032	DPTR 043	DPTR 071	ERRQ 038		

DPTR DESCRIPTIVE TEXT

ENTRY POINTS

- WRITE - AFTER A 1403 WRITE COMMAND HAS BEEN DECODED, ENTRY IS MADE TO THIS LABEL FROM THE -DPTC- ROUTINE. CARRIAGE DATA IS LOADED, AND THE PRINT LINE BUFFER IS LOADED IN SUB-SCAN SEQUENCE. POSITIONS OF THE PRINT LINE BUFFER FROM WHICH NO PRINTING IS DESIRED ARE SET TO ZERO. WHEN THE PLB COUNT AND THE UCW COUNT BOTH REACH ZERO, CHANNEL END IS SET.
- WXFER3 - ENTRY IS MADE HERE FROM THE -DPTS- ROUTINE WHEN DATA CHAINING IS IN EFFECT.
- CHEND - ENTRY IS MADE AT THIS LABEL FROM THE -DMCS- ROUTINE TO SET CHANNEL END AT THE COMPLETION OF AN MCS LOAD OPERATION.
- INCLN - WHEN A SENSE COMMAND IS ISSUED TO THE PRINTER AND DATA CHAINING HAD BEEN SPECIFIED, ENTRY IS MADE AT THIS LABEL FROM THE -DPTS- ROUTINE TO TEST THE SLI FLAG AND TO SET CHANNEL END.
- ENDOK - THIS IS THE NORMAL ENTRY FOR SETTING CHANNEL END FOR A SENSE COMMAND. ENTER FROM THE -DPTS- ROUTINE.
- CCBR - ENTRY IS MADE HERE FROM THE -EXFR- ROUTINE TO ATTEMPT A CHANNEL END INTERRUPT
- CARCTL - WHEN A CARRIAGE CONTROL OPERATION IS DECODED IN THE 1403 COMMAND DECODE ROUTINE -DPTC-, ENTRY IS MADE HERE TO SET THE CARRIAGE CONTROL REGISTER AND SET CHANNEL END.
- IMMEDI - A 1403 NO-OP COMMAND FROM -DPTC-, OR A READER COMMAND IMMEDIATE FROM -EXFR-, ENTER AT THIS LABEL TO SET THE SECONDARY BIT AND CHANNEL END.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DPTR 001	T			1403 WRITE-CHANNEL END, AND CARRIAGE CTRL ROUTINE	
		DPTR 003	*			*****	
		DPTR 004	*			*** WRITE COMMANDS ***	
		DPTR 005	*			*****	
OCDC	EB60	DPTR 006	WRITE	008	GLCAD	BR IF H16=0	BR IF NOT GATE LOAD
OCDE	2D20	DPTR 007				SET PRA K=02	BLOCK TWO HOME PULSES
OC E0	4DCF	DPTR 008	GLCAD			PRC=PO	LOAD CARRIAGE DATA
OC E2	2813	DPTR 009	RWBAL			IO=0\$K01	SET PLBAR COUNT 01 SET DE
OC E4	2985	DPTR 010				I1=0\$K80	
OC E6	294B	DPTR 011				I1=I1+K04	**SET PLB COUNT 84 DEC 132
OC E8	2A25	DPTR 012				T0=0\$K20	
OC EA	2ACB	DPTR 013				T0=T0+K0C	**SET PLBAR CONST 2C DEC 44
OC EC	2C75	DPTR 015				PO=0\$K70	PLB TABLE ADDR FOR NDT 24K
OC EE	5D38	DPTR 019	WXFER1			RDB P1 V+1	**READ CHAR FROM DATA ADDR
OC F0	5BC0	DPTR 020				RDB T1 AS,P+0	**READ CHAR FROM TABLE
OC F2	4F8F	DPTR 021				PR=IO	**LOAD PLBAR
OC F4	2D04	DPTR 022				SET PRA K=20	SET READ CONTROL
OC F6	0BFF	DPTR 023				Z=T10KFF	
OC F8	C4FC	DPTR 024		026	WXFER2	BR IF ZN	**BRANCH IF CHAR NOT FF
OC FA	3F83	DPTR 025				H1=H1\$K08	**SET DATA CHECK
OC FC	4BBF	DPTR 026	WXFER2			PRO=T1	**LOAD PLB

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
00FE	5006	DP TR 027				U=U-1	
0000	C488	DP TR 028		033	UCNTOW	BR IF Z=0	**BR IF UCW COUNT 0
0002	29FF	DP TR 029	WXFER3			I1=I1+KFF	**PLB COUNT MINUS 1
0004	C49F	DP TR 030		044	WRGLEN	BR IF Z=0	**BR IF PLB COUNT 0
0006	82E4	DP TR 031		093	CNT89	BAL	GO ADJUST PLBAR ADDRESS
0008	8CEE	DP TR 032		019	WXFER1	BR	**BR TO NEXT BYTE
000A	C56B	DP TR 033	UCNTOW	043	GOCHN	BR IF G1 BIT0=1	**BR IF DATA CHAIN FLAG
000C	29FF	DP TR 034	WXFER4			I1=I1+KFF	**PLB COUNT MINUS 1
000E	C4A1	DP TR 035		045	XFREND	BR IF Z=0	**BR IF PLB COUNT=0
0010	3445	DP TR 036				GO=GO\$K40	**SET INCORRECT LENGTH
0012	82E4	DP TR 037		093	CNT89	BAL	GO ADJUST PLBAR ADDRESS
0014	4F8F	DP TR 038	XFRBL			PR=IO	**LOAD PLBAR
0016	2D04	DP TR 039				SET PRA K=20	SET READ CONTROL
0018	2B05	DP TR 040				T1=0\$K00	SET PLB CHAR TO 00
001A	48BF	DP TR 041				PRO=T1	**LOADPLB
001C	8DOC	DP TR 042		034	WXFER4	BR	**BR TO NEXT BYTE
006A	9D44	DP TR 043	GOCHN	DPTQ 036	CHAIOK	BR	
001E	3445	DP TR 044	WRGLEN			GO=GO\$K40	**SET INCORRECT LENGTH
0020	EAA8	DP TR 045	XFREND	049	NPCHK1	BR IF PRS BIT6=0	BR IF NO PARITY CHECK
0022	3E23	DP TR 046				H0=H0\$K02	**SET UNIT CHECK
0024	3F15	DP TR 047				H1=H1\$K10	**SET EQUIP CHECK
0026	1545	DP TR 048				G1=G1*-K40	**RESET COMMAND CHAIN FLAG
0028	DB2C	DP TR 049	NPCHK1	051	CHKEND	BR IF H1 BIT5=0	**BR IF NO BLOCK DATA CHECK
002A	1F83	DP TR 050				H1=H1*-K08	**RESET DATA CHECK
002C	3D02	DP TR 051	CHKEND			SET PRA K=90	SET PRINT GATE AND BUSY
		DP TR 052	*	*****			
		DP TR 053	*	CHANNEL END ROUTINE			
002E	1F23	DP TR 054	CHEND			H1=H1*-K02	RST GATE LOAD
0030	D440	DP TR 055		064	ENDCK	BR IF GO BIT1=0	**BR IF NO INCORRECT LENGTH
0032	C538	DP TR 056		060	INCLEN	BR IF G10=0	BR IF NO DATA CHAIN
0034	40G6	DP TR 057				U=U	
0036	C4C0	DP TR 058		064	ENDCK	BR IF ZNZ	BR IF CCW COUNT NOT ZERO
0038	E53E	DP TR 060	INCLEN	063	NOSLI	BR IF G1 BIT2=0	**BR IFNC SLI FLAG
003A	1445	DP TR 061				GO=GO*-K40	**RESET INCORRECT LENGTH
003C	8D40	DP TR 062		064	ENDOK	BR	
003E	1545	DP TR 063	NOSLI			G1=G1*-K40	**RESET COMMAND CHAIN FLAG
0040	3E83	DP TR 064	ENDOK			H0=H0\$K08	**SET CHANNEL END
0042	3485	DP TR 065				GO=GO\$K80	**SET SECONDARY BIT
0044	D550	DP TR 066	CCBR	072	STUCW3	BR IF G11=0	** BR IF NO CHAIN--TRY FOR CHAN.END INT
0046	B106	DP TR 067	CMDCHN	DCOM 006	PCIBR	BAL **CHAIN IS ON ,GO CHECK PCI FLAG,TRY FOR PCI INT	
0048	1E83	DP TR 068				H0=H0*-K08	RST CH. END----FOR CHAINING
004A	DA52	DP TR 069		073	NCHAIN	BR IF H05=0	BR IF NO DEV END YET
004C	9258	DP TR 070		CCOM 066	LRSTRB	BAL	**BR & LINK TO RESTORE LS
004E	9D44	DP TR 071		DPTQ 036	CHAIOK	BR	
0050	9D52	DP TR 072	STUCW3	DPTQ 019	TAKEIN	BR	BR TO INTERRUPT ROUTINE
0052	9D54	DP TR 073	NCHAIN	DPTQ 020	STINST	BR **	
		DP TR 075	*	*****			
		DP TR 076	*	CARRIAGE CONTROL COMMAND			
0054	4DCF	DP TR 077	CARCTL			PRC=PO	**LOAD CARRIAGE DATA
0056	2D02	DP TR 078				SET PRA K=10	SET PRINT BUSY LATCH
0058	C55C	DP TR 080	IMMEDI	082	OKCHAI	BR IF G10=0	BR IF NOT CHAIN DATA
005A	15E5	DP TR 081				G1=G1*-KE0	RST CC,CD,SLI
005C	3485	DP TR 082	OKCHAI			GO=GO\$K80	SET SECONDARY BIT

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0D5E	3E83	DPTR 083				H0=H0\$K08	**SET CHANNEL END
0D60	0547	DPTR 084		067	CMDCHN	BR IF G1 BIT1=1**BR IF THIS COMMAND CHAIND	
0D62	F251	DPTR 085		072	STUCW3	BR IF D0 BIT7=1**BR IF LAST COMMAND CHAIND	
0D64	DA68	DPTR 086		088	NOTNOP	BR IF H05=0	BR IF NO DE
0D66	1485	DPTR 087				GO=GO*-K80	RST SECONDARY BIT
0D68	9D3C	DPTR 088	NOTNOP	DPTS 066	PCENTR	BR	
		DPTR 090	*			*****	
		DPTR 091	*			*****	
		DPTR 092	*			*****	
02E4	2DA5	DPTR 093	CNT89			P1=0\$KA0	SET UP
02E6	3D53	DPTR 094				P1=P1\$K09	CONSTANT OF 169
02E8	2BA5	DPTR 095				T1=0\$KA0	PUT CONSTANT OF
02EA	3B73	DPTR 096				T1=T1\$K07	167 IN T1
02EC	6883	DPTR 097				T1=T1+I0	ADD 167 TO PLBAR AND
02EE	F4F5	DPTR 098		101	GREA89	BR IF AC=1	BR IF PLBAR=89 OR GREATER
02F0	68A3	DPTR 099				I0=I0+T0	NOT 89 YET ADD 44
02F2	128E	DPTR 100				RTN	RETURN TO READ OR WRITE
02F4	68D3	DPTR 101	GREA89			I0=I0+P1	ADD 169 TO PLBAR
02F6	128E	DPTR 102				RTN	RETURN TO READ OR WRITE
		DPTR 103	*			*****	

 * CROSS REFERENCE FOR CSECT DPTR *

DPTR 006	DPTC 084	DPTC 092	
DPTR 008	DPTR 006		
DPTR 019	DPTR 032		
DPTR 026	DPTR 024		
DPTR 029	DPTS 063		
DPTR 033	DPTR 028		
DPTR 034	DPTR 042		
DPTR 043	DPTR 033		
DPTR 044	DPTR 030		
DPTR 045	DPTR 035		
DPTR 049	DPTR 045		
DPTR 051	DPTR 049		
DPTR 054	DMCS 052		
DPTR 060	DPTR 056	DPTS 094	
DPTR 063	DPTR 060		
DPTR 064	DPTR 055	DPTR 058	DPTR 062 DPTS 095
DPTR 066	EXFR 144		
DPTR 067	DPTR 084		
DPTR 072	DPTR 066	DPTR 085	
DPTR 073	DPTR 069		
DPTR 077	DPTC 086		
DPTR 080	DPTC 022	EXFR 131	
DPTR 082	DPTR 080		
DPTR 088	DPTR 086		
DPTR 093	DPTR 031	DPTR 037	
DPTR 101	DPTR 058		

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DPTS 031	*	3-PROT CHECK		3-SKIP	3-BUSY 3-EQUIP CHK
		DPTS 032	*	4-CH DATA CHK		4-PCI	4-CH END 4-DATA CHK
		DPTS 033	*	5-CH CTRL CHK		5-ACTIVE	5-DEV END 5-BLOK DAT CHK
		DPTS 034	*	6-INTFCE CHK		6-INPUT	6-UNIT CHK 6-GATE LOAD
		DPTS 035	*	7-INT IN BUFFER		7-OUTPUT	7-UNIT EXEC 7-CHAN 9
		DPTS 036	*				
		DPTS 037	*			*****	
		DPTS 038	*			START I/O ACTIVE-0 SECONDARY-0	
		DPTS 039	*			*****	
		DPTS 040	*				
		DPTS 041	*				
1CFC	2482	DPTS 042	PRSI0			SET MODE K=18	**SET 1403 MODE
1CFE	5449	DPTS 043				GO=GO	
1D00	C4B7	DPTS 044		062	CHDATA	BR IF Z=0	**BR IF DATA CHAIN
1D02	8DC4	DPTS 045		CCOM 058	LSAVED	BAL	**BR & LINK TO LS SAVE
1D04	54C9	DPTS 046				PO=GO	MOVE COMMAND TO PO
1D06	A078	DPTS 047		ERRQ 083	RDCHST	BAL	** GO GET CHAN. AND UNIT STATUS IN G,H
1D08	0C4B	DPTS 048				Z=PO=K04	LOOK FOR SENSE
1D0A	C497	DPTS 049		055	READY	BR IF Z=0	BR IF SENSE
1D0C	0C3B	DPTS 050				Z=PO=K03	LOOK FOR NO-OP
1D0E	C497	DPTS 051		055	READY	BR IF Z=0	BR IF NO-OP
1D10	1D00	DPTS 052				RST PRA K=80	RST EXT SENSE LATCH
1D12	1FF5	DPTS 053				H1=H1*-KFO	**CLEAR UNIT SENSE
1D14	1F93	DPTS 054				H1=H1*-K09	**REGEN BLOCK DATA CHECK AND GA.LOAD
1D16	2E05	DPTS 055	READY			HO=0\$K00	**CLEAR UNIT STATUS
1D18	DE9C	DPTS 056		058	NOTRDY	BR IF PRS BIT1=0	BR IF NOT READY
1D1A	BC05	DPTS 057		DPTC 029	COMHI N	N=POH	
1D1C	3F45	DPTS 058	NOTRDY			H1=H1\$K40	**SET INTERVENTION REQUIRED
1D1E	CC4B	DPTS 059				Z=PO=K04	
1D20	C4BA	DPTS 060		064	ISLCHK	BR IF ZNZ	**BR IF NO SENSE COMMAND
1D22	8F68	DPTS 061		084	SENSE	BR	
1D36	5460	DPTS 062	CHDATA			RDH G AS,D+0	GET CHANEL STATS AND FLAGS
1D38	8DC2	DPTS 063		DPTR 029	WXFER3	BR	
1D3A	2E23	DPTS 064	ISLCHK			HO=0\$K02	CLEAR UNIT STATUS, SET UC
1D3C	1547	DPTS 066	PCENTR			G1=G1*-K44	**RESET COMMAND CHAIN FLAG
1D3E	F224	DPTS 067		070	NOTINT	BR IF DO BIT7=0	** LAST COMM NOT CHAINED
1D40	3543	DPTS 068				G1=G1\$K04	TURN ON ACTIVE IF CHAINED
1D42	9D52	DPTS 069		DPTQ 019	TAKEIN	BR	**GO TO IMMED INTERRUPT
1D24	5E29	DPTS 070	NOTINT			VO=HO	SAVE UNIT STATUS
1D26	2EC7	DPTS 071				HO=0	NO UNIT STATUS IN UCW
1D28	A568	DPTS 072		DCOM 049	REDSTR	BAL	** STORE CH.ST,FLAGS AND UNIT ST.IN UCW
1D2A	52E9	DPTS 073				HO=VO	RESTORE UNIT STATUS
1D2C	C131	DPTS 074		077	NOPCIF	BR IF G14=1	BR IF PCI FLAG IS ON
1D2E	1485	DPTS 075				GO=GO*-K80	RST SECONDARY BIT
1D30	54F9	DPTS 077	NOPCIF			H1=GO	**MOVE CHANNEL STATUS
1D32	9258	DPTS 078		CCOM 066	LRSTRB	BAL	GET P AND I
1D34	A030	DPTS 079		CCOM 135	STATUS	BR	**BR TO STORE CSW HW#3
		DPTS 080	*			*****	
		DPTS 081	*		SENSE COMMAND		
		DPTS 082	*			*****	
UF66	1F45	DPTS 083	SENSE1			H1=H1*-K40	RST INTERVENTION REQ.
UF68	F573	DPTS 084	SENSE	089	SNSRTN	BR IF G13=1	BR IF SKIP FLAG ON
UF6A	5FB9	DPTS 085				T1=H1	**MOVE SENSE BYTE

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
OF6C	E271	DPTS 086		088	OVER	BR IF D06=1	SKIP NXT WORD IF 2540 SHARE
OF6E	1B63	DPTS 087				T1=T1*-K06	**STRIP SENSE BYTE
OF70	7B38	DPTS 088	OVER			STB T1 V+1	**STORE SENSE BYTE
OF72	2E43	DPTS 089	SNSRTN			HO=0\$K04	CLEAR UNIT STATUS, SET DE
OF74	5006	DPTS 090				U=U-1	**UCW COUNT MINUS 1
OF76	C57B	DPTS 091		093	FORGET	BR IF G10=1	BR IF DATA CHAIN-SET WLR BIT
OF78	C4FF	DPTS 092		095	SNSEND	BR IF Z=0	**BR IF UCW COUNT 0
OF7A	3445	DPTS 093	FORGET			GO=GO\$K40	**SET INCORRECT LENGTH
OF7C	8D38	DPTS 094		DPTR 060	INCLN	BR	**BR TO CHECK SLI FLAG
OF7E	8D40	DPTS 095	SNSEND	DPTR 064	ENDCK	BR	

 * CROSS REFERENCE FOR CSECT DPTS *

DPTS 042	DCLB 209		
DPTS 055	DPTS 049	DPTS 051	
DPTS 058	DPTS 056		
DPTS 062	DPTS 044		
DPTS 064	DPTC 025	DPTS 060	
DPTS 066	DPTR 088	ERDR 145	
DPTS 070	DPTS 067		
DPTS 077	DPTS 074	DYPE 150	
DPTS 083	DPTC 095		
DPTS 084	DPTS 061	DYPE 166	
DPTS 088	DPTS 086		
DPTS 089	DPTS 084		
DPTS 093	DPTS 091		
DPTS 095	DPTS 092		

DPTT DESCRIPTIVE TEXT

ENTRY POINTS

TEST00 - AFTER A TEST I/O OR A START I/O HAS BEEN DECODED IN THE -DCLA- ROUTINE, ENTRY IS MADE AT THIS LABEL TO TEST THE ACTIVE AND SECONDARY BITS TO DETERMINE FURTHER OPERATION.

UCENT - WHEN A NOT READY CONDITION IS DETECTED FOR THE PRINTER, READER-PUNCH, OR PR-KB, ENTRY IS MADE AT THIS LABEL TO SET UNIT CHECK AND ZERO OUT THE CHANNEL STATUS.

RESTES - THE 2540 REQUEST AND TEST I/O ROUTINE -ERRQ-, OR THE 1052 REQUEST ROUTINE -DYPE-, ENTER AT THIS LABEL TO CLEAR A CHANNEL END OR A CHANNEL END-DEVICE END CONDITION.

SHARE - ENTRY IS MADE AT THIS LABEL FROM -ERRQ-, OR -DYPE-. DEVICE END IS TESTED- IF DE ON - IF INTERRUPT HAS BEEN INITIATED, RESET INTERRUPT LATCH, AND STORE STATUS IN CSW. IF DE OFF - STATUS WITH BUSY SET IS STORED IN THE CSW.

DEVE - ENTRY IS MADE AT THIS LABEL FROM -DYPE- BECAUSE THE SECONDARY BIT IS ON AND THE ATTENTION BIT IS ON DURING INITIAL SELECTION FOR START I/O OR TEST I/O. RESET INTERRUPT IF ON, AND STORE STATUS IN CSW.

SENCHK - THE 1403 INTERRUPT REQUEST ROUTINE -DPTQ- ENTERS AT THIS LABEL TO CHECK THE PRINT SENSE DATA.

RESINB - THE CHANNEL 0 INTERRUPT ROUTINE -DCHN- ENTERS AT THIS LABEL TO RESET INTERRUPT, SECONDARY BIT, AND S7. RETURN TO -DCHN- TO STORE STATUS.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DPTT 001	T			1403 TEST I/O---START I/O WITH SEC.ON	
		DPTT 003	*			*****	
		DPTT 004	*			TEST I/O ACTIVE=0 SECONDARY=0	
		DPTT 005	*			*****	
0948	2482	DPTT 006	TEST00			SET MODE K=18	SET 1403 MODE
094A	A078	DPTT 007		ERRQ 083	RDCHST	BAL ** GO GET CHAN. AND UNIT STATUS IN G,H	
094C	D15B	DPTT 008		020	TEST11	BR IF G15=1	BR IF ACTIVE TEST IOO
094E	C421	DPTT 009		046	STTE01	BR IF G00=1	BR ON SEC TO START OR TEST NO A
0950	DED4	DPTT 010		013	UCENT	BR IF PRS1=0	BR IF NOT READY
0952	B362	DPTT 011		CCOM 154	CCOB	BR ** GO TO COMM MODE RST,CC 0 SET	
0954	2E23	DPTT 013	UCENT			H0=0\$K02	SET UNIT CHECK IN STA
0956	2F07	DPTT 014				H1=0	ZERO OUT CHANNEL STATUS
0958	AC1E	DPTT 015		CCOM 123	STATOS	BR ** STORE 0'S IN CSW WITH NEW STATUS	
		DPTT 016	*			*****	
		DPTT 017	*			TEST I/O ACTIVE=1 SECONDARY=1	
		DPTT 018	*			*****	
095A	0D08	DPTT 020	TEST11			RST PRA K=40	RST PRINT REQUEST
095C	5FB9	DPTT 021				T1=H1	SAVE SENSE BYTE
095E	E401	DPTT 023	RESTES	026	PROGF 0	BR IF G02=1	BR IF PROGRAM ERROR
0960	AA03	DPTT 024		026	PROGF N	BR IF H0=NZ	BR IF NO STATUS
0962	AC74	DPTT 025		CCOM 161	CC2B	BR	GO SET BUSY

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0900	154B	DPTT 026	PROGF 0			G1=G10K04	RESET ACTIVE BIT
0902	B31E	DPTT 027		088	RSTINT	BAL	GO RST S7,INT BUFF BIT,SEC.
0904	DA0F	DPTT 028		033	ZSECON	BR IF H05=1	BR IF DE
0906	CA0E	DPTT 029		033	ZSECON	BR IF H04=0	BR IF NO CHEND OR NO DEV.END
0908	E40F	DPTT 030		033	ZSECON	BR IF G02=1	BR IF PRGRAM CHK-DONT SET SEC
090A	2C85	DPTT 031				U0=0\$K80	STORE SEC FOR CHE ALONE-UCW
090C	8910	DPTT 032		034	UCWSTR	BR	
090E	2007	DPTT 033	ZSECON			U0=0	DE,CHE STAT-CLEAR CH STAT IN UCW
0910	7068	DPTT 034	UCWSTR			STH U AS,D+2	UPDATE UCW CH STAT AND FLAGS
0912	C116	DPTT 035		037	DONTCH	BR IF G14=0	BR IF NO PCI FLAG
0914	3485	DPTT 036				GO=G0\$K80	SET PCI IN CHAN STAT
0916	54F9	DPTT 037	DONTCH			H1=GO	MOVE CHANNEL STATUS
0918	7A68	DPTT 038				STH T AS,D+2	D=COCC
091A	5068	DPTT 039				RDH U AS,D+2	UCW COUNT=U
091C	5260	DPTT 040				RDH V AS,D+0	CCW ADDRESS=V
091E	A00E	DPTT 041	CALLH	CCOM 110	FULCSW	BR ** STORE CSW RST MODE, SET CC1	
		DPTT 042	*			*****	
		DPTT 043	*			TEST OR START I/O ACTIVE=0 SECONDARY=1	
		DPTT 044	*			*****	
0920	5E29	DPTT 046	STTE01			V0=H0	MOVE UNIT STATUS
0922	CEB1	DPTT 047		056	EXTDE	BR IF PRS0=1	BR IF EXT DE
0924	DA39	DPTT 049	SHARE	061	DEVE	BR IF H0 BIT5=1	BR IF DE.
0926	2E15	DPTT 050				H0=0\$K10	SET BUSY
0928	54F9	DPTT 051				H1=GO	MOVE CHANNEL STATUS
092A	1F85	DPTT 052				H1=H1*-K80	RST SECONDARY
092C	F246	DPTT 053		068	DELIC	BR IF D07=0	BR IF START-STORE STATUS ONLY
092E	A01E	DPTT 054	GOSTOR	CCOM 123	STATOS	BR	THIS IS TEST STORE STAT AND 0,S
0930	B14C	DPTT 056	EXTDE	071	SENCHK	BAL	GO CHECK SENSE
0932	5F39	DPTT 057				V1=H1	PUT UPDATED SENSE IN V1
0934	6664	DPTT 058				D=D+2	UPDATE UCW ADD TO STATUS-CA
0936	726A	DPTT 059				STH V AS,D-2	STORE UNIT STAT
0938	B31E	DPTT 061	DEVE	088	RSTINT	BAL ** GO RST S7,INT BUFF. BIT AND SEC.BIT	
093A	7460	DPTT 062				STH G AS,D+0	UCW CH STATUS
093C	54F9	DPTT 063				H1=GO	MOVE CH STATUS
093E	E243	DPTT 064		066	NOTPTR	BR IF D06=1	BR IF NOT PRINTER
0940	0D08	DPTT 065				RST PRA K=40	RST PRINTER REQUEST
0942	F22F	DPTT 066	NOTPTR	054	GOSTOR	BR IF D07=1	BR IF TEST I/O
0944	3E15	DPTT 067				H0=H0\$K10	SET BUSY
0946	A030	DPTT 068	DELIC	CCOM 135	STATUS	BR	STORE STATUS
		DPTT 069	*			*****	
314C	FED0	DPTT 071	SENCHK	073	CHAN9	BR IF PRS3=0	BR IF NO CHANNEL 12
314E	3E13	DPTT 072				H0=H0\$K01	SET UNIT EXCEPTION
3150	EED6	DPTT 073	CHAN9	076	DATACH	BR IF PRS2=0	BR IF NO CHANNEL 9
3152	3E23	DPTT 074				H0=H0\$K02	SET UNIT CHECK
3154	3F13	DPTT 075				H1=H1\$K01	SET CHANNEL 9 IN SENSE
3156	CB5A	DPTT 076	DATACH	078	HAMMER	BR IF H14=0	BR IF NO DATA CHECK
3158	3E23	DPTT 077				H0=H0\$K02	SET UNIT CHECK
315A	5E3F	DPTT 078	HAMMER			V1=PRS	MOVE SENSE
315C	1393	DPTT 079				V1=V1*-K09	LOOK FOR PARITY OR HAMMER CHECK
315E	FCE5	DPTT 080		083	SETDE	BR IF LZ=0	BR IF NO ERROR
3160	3E23	DPTT 081				H0=H0\$K02	SET UNIT CHECK
3162	3F15	DPTT 082				H1=H1\$K10	SET EQUIPMENT CHK IN SENSE
3164	3E43	DPTT 083	SETDE			H0=H0\$K04	SET DEVICE END

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
3166	GDOA	DPTT 084				RST PRA K=50	RST BUSY AND REQUEST
3168	128E	DPTT 085				RTN	GO BACK TO TEST I/O OR REQ
		DPTT 087	*			*****	
331E	FC24	DPTT 088	RSTINT	091	DEVEND	BR IF G07=0	BR IF NO INT IN BUFF
3320	1413	DPTT 089	RESINB			GO=GO*-K01	RESET INT INDICATOR
3322	0010	DPTT 090				RST S7	RESET INT LATCH
3324	4046	DPTT 091	DEVEND			U=G	MOVE CHAN STAT+FLAGS
3326	1485	DPTT 092				GO=GO*-K80	RST SEC FOR CSW STORE
3328	128E	DPTT 093				RTN	
		DPTT 094	*			*****	

 * CROSS REFERENCE FOR CSECT DPTT *

DPTT 006	DCLA 303						
DPTT 013	DPTT 010	DYPE 203	ERRQ 014				
DPTT 020	DPTT 008						
DPTT 023	DYPE 204	ERRQ 052					
DPTT 026	DPTT 023	DPTT 024					
DPTT 033	DPTT 028	DPTT 029	DPTT 030				
DPTT 034	DPTT 032						
DPTT 037	DPTT 035						
DPTT 046	DPTT 009						
DPTT 049	DYPE 207	ERRQ 053					
DPTT 054	DPTT 066						
DPTT 056	DPTT 047						
DPTT 061	DPTT 049	DYPE 208					
DPTT 066	DPTT 064						
DPTT 068	DPTT 053						
DPTT 071	DPTQ 026	DPTT 056					
DPTT 073	DPTT 071						
DPTT 076	DPTT 073						
DPTT 078	DPTT 076						
DPTT 083	DPTT 080						
DPTT 088	DPTT 027	DPTT 061					
DPTT 089	DCHN 045						
DPTT 091	DPTT 088						

DYPE DESCRIPTIVE TEXT

ADDITIONAL INFORMATION REFERENCE
2025 PROCESSING UNIT, F.E. THEORY OF OPERATION MANUAL

1052 UCW

AUX STORAGE ADDRESS----	0XF7	0XF8	0XF9	0XFA	0XFB	0XFC	0XFD	0XFE	0XFF
*****	*****	*****	*****	*****	*****	*****	*****	*****	*****
* SENSE *	* CHANNEL *	* FLAGS *	* DATA ADDRESS *	* COUNT *	* FIELD *	* NEXT CCW ADDRESS *			
* BYTE *	* STATUS *	* AND OP *	* UNIT *						
*****	*****	*****	*****	*****	*****	*****	*****	*****	*****
L.S. REG----	P1	GO	G1	HO STATUS ADDRESS VO-V1 DATA			UO-U1		

SENSE BYTE

- 0=COMMAND REJECT
- 1=INTERVENTION REQUIRED
- 2=DOES NOT APPLY
- 3=EQUIPMENT CHECK
- 4=DOES NOT APPLY
- 5=DOES NOT APPLY
- 6=DOES NOT APPLY
- 7=DOES NOT APPLY

CHANNEL STATUS

- 0=SECONDARY BIT
- 1=INCORRECT LENGTH
- 2=PROGRAM CHECK
- 3=PROTECTION CHECK
- 4=0 COUNT BIT
- 5=CHAN CONTROL CHECK
- 6=INTERFACE CHECK
- 7=PR-KB INTERRUPT IN INTERRUPT BUFFER

FLAGS AND OP

- 0=CHAIN DATA (CD) FLAG
- 1=CHAIN COMMAND (CC) FLAG
- 2=SUPPRESS LENGTH INDICATION (SLI) FLAG
- 3=SKIP FLAG
- 4=PROGRAM CONTROLLED INTERRUPT (PCI) FLAG
- 5=ACTIVE BIT
- 6-7=OP -- 01=WRITE
- 10=READ OR SENSE
- 11=WRITE WITH ACR

UNIT STATUS

- 0=ATTENTION
- 1=DOES NOT APPLY
- 2=DOES NOT APPLY
- 3=BUSY
- 4=CHANNEL-END
- 5=DEVICE-END
- 6=UNIT-CHECK
- 7=UNIT EXCEPTION

DYPE ENTRY POINTS

TWRREQ -- FROM -BSWI- TO HANDLE A SHARE REQUEST FOR ONE OF THE FOLLOWING.

1. ATTENTION	5. NOT READY TO READY
2. READ SHARE REQUEST	6. LOGOUT
3. WRITE SHARE REQUEST	7. ALTER/DISPLAY
4. INTERRUPT STKD REQ	

THE FOLLOWING BITS ARE TESTED AS NECESSARY TO DETERMINE THE STATUS OF THE CURRENT OPERATION.

- G1 BIT 5 -- ACTIVE BIT
- GO BIT 0 -- SECONDARY BIT
- PO BIT 4 -- ENDING STATUS BIT

GOCL -- FROM 2540 ROUTINE TO SHARE WORD THAT MOVES SENSE REG. AFTER THE WORD IS EXECUTED, THE -DYPE- ROUTINE BRANCHES TO -DPTS- TO DO THE SENSE OPERATION FOR EITHER THE 1052 OR 2540.

TWSEC7 -- FROM I/O INSTRUCTIONS ROUTINE -DCLA- FOR THE FOLLOWING CONDITIONS.

- START I/O -- ACT BIT=0, SEC BIT=1
- TEST I/O -- ACT BIT=1, SEC BIT=1, CC FLAG=0
- TEST I/O -- ACT BIT=0

OUTSTANDING STATUS IS STORED IN THE CSW.

NOATEN -- FROM WRAP TRAP ROUTINE. THIS ENTRY IS USED TO TRY FOR AN INTERRUPT.

HLTTW -- HALT I/O ENTRY FROM I/O INSTRUCTIONS ROUTINE, -DCLA-. THE ACTIVE AND SECONDARY BITS ARE TESTED TO DETERMINE THE NECESSARY ACTION AND CONDITION CODE SETTING.

TWOFF7 -- START I/O ENTRY. CCW FETCH HAS JUST BEEN COMPLETED IN -DCLB-. READY IS TESTED, THEN THE ROUTINE INITIALIZES FOR THE CURRENT COMMAND.

STTRSL -- TRANSLATE SUBROUTINE ENTRY. TRANSLATES KEYBOARD CODE CHARACTERS TO EBCDIC. USED BY ALTER/DISPLAY ROUTINE, AS WELL AS DURING A READ REQUEST CYCLE.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DYPE 001	T			1052 REQUEST ROUTINES	
1F10	FF8F	DYPE 002	TWRREQ	020	ADPACT	BR IF TU3=1	BR IF ALTER-DISPLAY ACTIVE
1F12	FA8D	DYPE 003		021	LOGREQ	BR IF TT7=1	BR IF LOGOUT REQUEST
1F14	8DC4	DYPE 004		CCOM 058	LSAVEB	BAL	GO SAVE P AND I
1F16	9C28	DYPE 005		168	GETUCW	BAL	GET UCW AND SET PR-KB MODE
1F18	C832	DYPE 006		022	NOBUFR	BR IF P04=0	BR IF NOT END STATUS
1F1A	CEA0	DYPE 007	AFTER	010	NOATEN	BR IF T10=0	BR IF ATTENTION LATCH IS OFF
1F1C	3E85	DYPE 008	ATTEN			H0=H0\$K80	SET ATTEN IN UNIT STATUS REG
1F1E	0F10	DYPE 009				RST TA K=01	RESET ATTEN LATCH
1F20	1F28	DYPE 010	NOATEN			RST TA K=C2	RST READ, WRITE, AND SHARE REQ
1F22	3C83	DYPE 011				PO=PO\$K08	TURN ON END STATUS PRESENT
1F24	B10A	DYPE 012		DCOM 008	INTRPT	BAL	TRY FOR NORMAL INTRPT (NOT PCI)
1F26	CEAA	DYPE 013		015	QATND	BR IF T10=0	BR IF ATTENTION LATCH NOT SET
1F28	2F04	DYPE 014				SET TA K=20	SET MICROFORCE LATCH
1F2A	9A90	DYPE 015	QATND	180	PUTUCS	BAL	MOVE IN NEW UNIT STATUS REG
		DYPE 016	*				STORE UCW
1F2C	FEB1	DYPE 017		019	ADPREQ	BR IF TT3=1	BR IF ALTER-DISPLAY LATCH ON
1F2E	AAA2	DYPE 018		BSWI 066	RESTRH	BR	RETURN FROM REQUEST
1F30	961C	DYPE 019	ADPREQ	ALDP 022	START	BR	GO TO ALTER-DISPLAY START
1F0E	A6B8	DYPE 020	ADPACT	ALDP 216	RTTYP	BR	ALDP REQUEST, GO TYPE CHAR
1F0C	937A	DYPE 021	LOGREQ	BMCK 047	LOGOUT	BR	GO HANDLE LOGOUT REQUEST
1F32	D149	DYPE 022	NOBUFR	034	AC1	BR IF G15=1	BR IF ACTIVE BIT ON
1F34	CE39	DYPE 023		025	QATYES	BR IF H00=1	BR IF ATTN BIT IN U-STATUS REG
1F36	2E43	DYPE 024				H0=0\$K04	SET DEVICE END IN U-STATUS REG
1F38	C421	DYPE 025	QATYES	010	NOATEN	BR IF G00=1	BR IF SECONDARY BIT ON
		DYPE 026	*		ACTIVE 0	SECONDARY 0	
1F3A	2485	DYPE 027				GO=0\$K80	RST CH STATUS REG, SET SEC BIT
1F3C	DEA1	DYPE 028		010	NOATEN	BR IF TT1=1	BR IF NOT-READY TO READY
1F3E	2EC7	DYPE 029				H0=0	CLEAR UNIT STATUS REG
1F40	CE9D	DYPE 030		008	ATTEN	BR IF T10=1	BR IF ATTENTION LATCH SET
1F42	FEB1	DYPE 031		019	ADPREQ	BR IF TT3=1	BR IF ALTER-DISPLAY LATCH ON
1F44	3423	DYPE 032				GO=G0\$K02	SET INTERFACE CHECK
1F46	9F20	DYPE 033		010	NOATEN	BR	
1F48	C47D	DYPE 034	AC1	062	OSNS	BR IF G00=1	BR IF SECONDARY BIT ON
		DYPE 035	*		ACTIVE 1	SECONDARY 0	
1F4A	2E07	DYPE 036				H0=0	CLEAR UNIT STATUS REG
1F4C	EED2	DYPE 037		040	SINTVN	BR IF TT2=0	BR NOT INTERVENTION REQUIRED
1F4E	3D45	DYPE 038				P1=P1\$K40	SET INTVN REQ IN SENSE REG
1F50	BOCC	DYPE 039	DOCRLF	071	CRLF	BR	
1F52	B106	DYPE 040	SINTVN	DCOM 006	PCIBR	BAL	IF PCI, TRY FOR PCI INTERRUPT
1F54	DF8A	DYPE 041		066	READOT	BR IF TU1=0	BR IF READ OPERATION
		DYPE 042	*		WRITE OPERATION		
1F56	C06B	DYPE 043		053	WROBIT	BR IF G04=1	BR IF 0-COUNT BIT SET
1F58	5F38	DYPE 044				RDB H1 V+1	READ CHAR AT DATA ADDRESS
1F5A	4FFF	DYPE 045				TE=H1	SEND CHAR TO TE DATA REG
1F5C	0F20	DYPE 046				RST TA K=02	ISSUE SHARE RESET
1F5E	5006	DYPE 047	WCOMOT			U=U-1	DECREMENT COUNT
1F60	C4E6	DYPE 048		051	MORE	BR IF ZNZ	BR IF COUNT = 0
1F62	C505	DYPE 049		067	CHAIN1	BR IF G10=1	BR IF CD FLAG
1F64	3483	DYPE 050				GO=G0\$K08	SET 0 COUNT BIT
1F66	9A92	DYPE 051	MORE	181	PUTUCW	BAL	STORE UCW EXCEPT UNIT STATUS
1F68	AAA2	DYPE 052		BSWI 066	RESTRH	BR	RETURN FROM REQUEST
1F6A	1483	DYPE 053	WROBIT			GO=G0*-K08	RESET 0 COUNT BIT

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1F6C	E573	DYPE 054	ADIM	057	SLION	BR IF G12=1	BR IF SLI FLAG ON
1F6E	3445	DYPE 055				GO=GO\$K40	SET WLR
1F70	15C5	DYPE 056				G1=G1*-KCO	BREAK CHAIN (RESET CC,CD FLAGS)
1F72	E151	DYPE 057	SLION	039	DOCRLF	BR IF G16=1	BR IF WRITE ACR OR READ OP
1F74	5DD9	DYPE 058				P1=P1	
1F76	C4FD	DYPE 059		062	OSNS	BR IF Z=0	BR IF SENSE REG IS ZERO
1F78	15C5	DYPE 060				G1=G1*-KCO	RESET CC, CD FLAGS
1F7A	3E23	DYPE 061				HO=HO\$K02	SET UNIT CHECK
1F7C	D505	DYPE 062	OSNS	067	CHAIN1	BR IF G11=1	BR IF CC FLAG
1F7E	3EC3	DYPE 063				HO=HO\$K0C	SET CHANNEL END, DEVICE END
1F80	3485	DYPE 064	QFLIP			GO=GO\$K80	SET SECONDARY BIT
1F82	9F20	DYPE 065		010	NOATEN	BR	
1F0A	A2CA	DYPE 066	READOT	089	READ	BR	
1F04	1485	DYPE 067	CHAIN1			GO=GO*-K80	RESET SECONDARY BIT
1F06	9A90	DYPE 068		180	PUTUCS	BAL	MOVE IN NEW UNIT STATUS REG
		DYPE 069	*				STORE UCW
1F08	A49E	DYPE 070		154	KKKKKK	BR	GO RESTORE P, I, THEN CHAIN
30CC	2F53	DYPE 071	CRLF			H1=0\$K05	SET UP
30CE	3F15	DYPE 072				H1=H1\$K10	NEW LINE CHARACTER
30D0	1F00	DYPE 073				RST TA K=80	RESET READ LATCH
30D2	2F08	DYPE 074				SET TA K=40	SET WRITE LATCH
30D4	4FFF	DYPE 075				TE=H1	SEND NL CHAR TO TE DATA REG
30D6	CF20	DYPE 076				RST TA K=02	ISSUE SHARE RESET
30D8	3485	DYPE 077				GO=GO\$K80	SET SECONDARY BIT
30DA	1483	DYPE 078				GO=GO*-K08	RESET O-COUNT BIT
30DC	5DD9	DYPE 079				P1=P1	
30DE	C4E5	DYPE 080		083	OKDE	BR IF Z=0	BR IF SENSE REG IS ZERO
30E0	3E23	DYPE 081				HO=HO\$K02	SET UNIT CHECK
30E2	15C5	DYPE 082				G1=G1*-KCO	RESET CC, CD FLAGS
30E4	D56B	DYPE 083	OKDE	086	EXITA	BR IF G11=1	BR IF CC FLAG
30E6	3E83	DYPE 084				HO=HO\$K08	SET CHANNEL END
30E8	B10A	DYPE 085		DCOM 008	INTRPT	BAL	TRY FOR NORMAL INTRP (NOT PCI)
30EA	9A90	DYPE 086	EXITA	180	PUTUCS	BAL	GET NEW UNIT STATUS, STORE UCW
30EC	AAA2	DYPE 087		BSWI 066	RESTRH	BR	RETURN FROM REQUEST
		DYPE 088	*		READ OPERATION		
22CA	5AFF	DYPE 089	READ			H1=TI	GET KB CHAR FROM TI REG
22CC	1FC5	DYPE 090				H1=H1*-KCO	RESET CASE BITS 0,1
22CE	DAE9	DYPE 091		102	ALTCOD	BR IF TT5=1	BR IF ALTERNATE CODING
22D0	CAD4	DYPE 092		094	ARNSET	BR IF TT4=0	BR IF NOT KEYBOARD CHECK
22D2	2D15	DYPE 093	SETEQP			P1=0\$K10	CLEAR SENSE REG, SET EQUIP CHK
22D4	C065	DYPE 094	ARNSET	111	EXITB	BR IF G04=1	BR IF O-COUNT BIT SET
22D6	4806	DYPE 095				I=U	SAVE COUNT
22D8	B17A	DYPE 096		231	STTRSL	BAL	XLATE KB CHAR AND PRINT IT
22DA	4C86	DYPE 097				U=I	RESTORE COUNT
22DC	42CF	DYPE 098				STPO=PO	SET PROTECT KEY
22DE	F563	DYPE 099		101	SKIP	BR IF G13=1	BR IF SKIP FLAG
22E0	7F38	DYPE 100				STB H1 V+1	STORE CHAR READ AT DATA ADDR
22E2	9F5E	DYPE 101	SKIP	047	WCCMOT	BR	GO DECREMENT COUNT
22E8	0F5B	DYPE 102	ALTCOD			Z=H1\$K05	
22EA	C4F5	DYPE 103		108	RQNOF	BR IF Z=0	BR IF EOB
22EC	0FAB	DYPE 104				Z=H1\$K0A	
22EE	C4D2	DYPE 105		093	SETEQP	BR IF ZNZ	BR IF NOT CANCEL
22F0	2E13	DYPE 106				HO=0\$K01	SET UNIT EXCEPTION

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
22F2	1545	DYPE 107				G1=G1*-K40	RESET CC FLAG
22F4	CC67	DYPE 108	RQNOF	110	AD1MA	BR IF G04=1	BR IF 0-COUNT BIT IS SET
22F6	9F6C	DYPE 109		054	AD1M	BR	
22E6	B0CC	DYPE 110	AD1MA	071	CRLF	BR	
22E4	9F6A	DYPE 111	EXITB	053	WROBIT	BR	
		DYPE 112	*		START	I/O ACTIVE 0 SECONDARY 0	
24A2	8DC4	DYPE 113	TWOFF7	CCOM 058	LSAVEB	BAL	SAVE P AND I
24A4	54F9	DYPE 114	TWOFF8			H1=GO	MOVE COMMAND CODE
24A6	6664	DYPE 115				D=D+2	ADJUST UCW ADR REG TO DATA ADR
24A8	9A98	DYPE 116		184	PUTUCA	BAL	STORE DATA ADR AND COUNT IN UCW
24AA	9028	DYPE 117		168	GETUCW	BAL	GET UCW, SENSE BYTE, SET MODE
24AC	1485	DYPE 118				GO=GO*-K80	RESET SECONDARY BIT
24AE	1C83	DYPE 119				PO=PO*-K08	RESET ENDING STATUS BIT
24B0	9A92	DYPE 120		181	PUTUCW	BAL	STORE UCW EXCEPT UNIT STATUS
24B2	2E07	DYPE 121				HO=0	CLEAR UNIT STATUS REG
24B4	5FF9	DYPE 122				H1=H1	
24B6	C4BA	DYPE 123		125	CONTUE	BR IF ZNZ	BR IF NOT CHAIN DATA
24B8	AAA2	DYPE 124		BSWI 066	RESTRH	BR	GO RESTORE CPU MODE
24BA	1D45	DYPE 125	CONTUE			P1=P1*-K40	RESET INTERVENTION REQUIRED BIT
24BC	1F28	DYPE 126				RST TA K=C2	RST READ, WRITE, AND SHARE REQ
24BE	EEC2	DYPE 127		129	NONRDY	BR IF TT2=0	BR IF READY
24C0	3D45	DYPE 128				P1=P1\$K40	SET INTERVENTION REQUIRED
24C2	0F4B	DYPE 129	NONRDY			Z=H1\$K04	
24C4	C4E5	DYPE 130		163	SENSE	BR IF Z=0	BR IF SENSE COMMAND
24C6	EEDD	DYPE 131		142	NOTRDY	BR IF TT2=1	BR IF NOT READY
24C8	0F3B	DYPE 132				Z=H1\$K03	
24CA	C499	DYPE 133		151	NOOPRA	BR IF Z=0	BR IF NO-OP COMMAND
24CC	2DC7	DYPE 134				P1=0	CLEAR SENSE REG
24CE	0F1B	DYPE 135				Z=H1\$K01	
24D0	C495	DYPE 136		157	WRIT	BR IF Z=0	BR IF WRITE COMMAND
24D2	GF9B	DYPE 137				Z=H1\$K09	
24D4	C493	DYPE 138		156	WRTACR	BR IF Z=0	BR IF WRITE ACR COMMAND
24D6	0FAB	DYPE 139				Z=H1\$K0A	
24D8	C48B	DYPE 140		159	READOP	BR IF Z=0	BR IF READ COMMAND
24DA	3D85	DYPE 141				P1=P1\$K80	SET COMMAND REJECT IN SENSE REG
24DC	2E23	DYPE 142	NOTRDY			HO=0\$K02	SET UNIT CHECK IN U-STATUS REG
24DE	15C5	DYPE 143				G1=G1*-KC0	RESET CC, CD IN FLAGS/OP REG
24E0	F200	DYPE 144	NOTCCB	146	NOTCCA	BR IF D07=0	BR NOT CHAINED FROM PRIOR CCW
24E2	9F80	DYPE 145		064	QFLIP	BR	GO SET SEC BIT, TRY FOR INTRPT
2480	1547	DYPE 146	NOTCCA			G1=G1*-K44	RESET CC FLAG AND ACTIVE BIT
2482	9A90	DYPE 147		180	PUTUCS	BAL	GET NEW UNIT STATUS, STORE UCW
2484	C1G8	DYPE 148		150	ARPCI	BR IF G14=0	BR IF NO PCI FLAG
2486	3485	DYPE 149				GO=GO\$K80	SET SEC BIT IN CH STATUS REG
2488	9D30	DYPE 150	ARPCI	DPTS 077	NOPCIF	BR	GO MOVE STATUS AND STORE IN CSW
2498	2EC3	DYPE 151	NOOPRA			HO=0\$K0C	SET CHANNEL END, DEVICE END
249A	D560	DYPE 152		144	NOTCCB	BR IF G11=0	BR IF NO CC FLAG
249C	A566	DYPE 153		DCOM 048	CHANOK	BAL	STORE UCW, RESTORE P I, CHAIN
249E	9258	DYPE 154	KKKKKK	CCOM 066	LRSTRB	BAL	GO RESTORE P I, THEN CHAIN
24A0	93CA	DYPE 155		DCLB 039	CHAIN	BR	GO GET NEXT CCW FOR CHAINING
2492	3523	DYPE 156	WRTACR			G1=G1\$K02	SET BIT 6 FLAGS/OP REG
2494	2F08	DYPE 157	WRIT			SET TA K=40	SET WRITE LATCH
2496	9F66	DYPE 158		051	MORE	BR	GO STORE UCW, RESTORE CPU MODE
248A	3F20	DYPE 159	READOP			SET TA K=82	SET READ, INITIAL PRTR LATCHES

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
248C	EB90	DYPE 160		162	INNN	BR IF TU6=0	BR IF INITIAL PRTR NOT SET
248E	4FFF	DYPE 161				TE=H1	SEND CHARACTER TO 1052 DATA REG
2490	9F66	DYPE 162	INNN	051	MORE	BR	GO STORE UCW, RESTORE CPU MODE
24E4	3C83	DYPE 163	SENSE			PO=PO\$K08	SET ENDING STATUS BIT
24E6	5A92	DYPE 164		181	PUTUCW	BAL	STORE UCW EXCEPT UNIT STATUS
24E8	5DF9	DYPE 165	GOCAL			H1=P1	MOVE SENSE REG
24EA	8F68	DYPE 166		DPTS 084	SENSE	BR	GO SHARE PRINTER SENSE ROUTINE
		DYPE 167	*	UCW	FETCH		
1028	2791	DYPE 168	GETUCW			D1=0-K09	SET AUX STORAGE
102A	566D	DYPE 169				DO=D0L	ADDRESS
102C	16E3	DYPE 170				DO=DO*-K0E	0XF6
102E	24C4	DYPE 171				SET MODE K=20	SET PR-KB MODE
1030	5C68	DYPE 172				RDH P AS,D+2	GET PROTECT KEY, SENSE BYTE
1032	5468	DYPE 173				RDH G AS,D+2	READ CHNL STATUS, FLAGS/OP
1034	5268	DYPE 174				RDH V AS,D+2	DATA ADDRESS/UNIT STATUS
1036	5068	DYPE 175				RDH U AS,D+2	COUNT FIELD
1038	52E9	DYPE 176				HO=V0	MOVE UNIT STATUS
103A	2771	DYPE 177				D1=0-K07	RESTORE UCW ADDRESS TO F8
103C	128E	DYPE 178				RTN	
		DYPE 179	*				UCW STORE
1A90	5E29	DYPE 180	PUTUCS			V0=H0	MOVE UNIT STATUS
1A92	2791	DYPE 181	PUTUCW			D1=0-K09	SET ADDRESS TO 0XF6
1A94	7C68	DYPE 182				STH P AS,D+2	STORE SENSE BYTE
1A96	7468	DYPE 183				STH G AS,D+2	CHANNEL STATUS, FLAGS/OP
1A98	7268	DYPE 184	PUTUCA			STH V AS,D+2	DATA ADDRESS/UNIT STATUS
1A9A	7068	DYPE 185				STH U AS,D+2	COUNT FIELD
1A9C	2771	DYPE 186				D1=0-K07	RESTORE UCW ADDRESS TO F8
1A9E	128E	DYPE 187				RTN	
		DYPE 188	*				* START I/O ACT=0, SEC=1 (D07=0)
		DYPE 189	*				* TEST I/O ACT=1, SEC=1, CC FLAG=0
		DYPE 190	*				* TEST I/O ACTIVE=0
16F4	2791	DYPE 191	TWSECT			D1=0-K09	SET ADDRESS TO 0XF6
16F6	5860	DYPE 192				RDB T1 AS,D	READ SENSE BYTE
16F8	58BB	DYPE 193				T1=T1H	RESET SENSE BITS 4-7
16FA	7860	DYPE 194				STB T1 AS,D	STORE SENSE BYTE
16FC	6664	DYPE 195				D=D+2	INCREMENT UCW ADDRESS TO F8
16FE	2404	DYPE 196				SET MODE K=20	SET PR-KB MODE
1700	A078	DYPE 197		ERRQ 083	RDCHST	BAL	GO READ CHANNEL AND UNIT STATUS
1702	0F04	DYPE 198				RST TA K=20	RESET MICROFORCE LATCH
1704	D111	DYPE 199		204	ZA1S1	BR IF G15=1	BR IF ACTIVE BIT IS ON
1706	C413	DYPE 200		205	ZA0S1	BR IF G00=1	BR IF SECONDARY BIT IS ON
1708	EE8D	DYPE 201		203	TNRCNT	BR IF TT2=1	BR IF INTERVENTION REQUIRED
170A	B362	DYPE 202		CCOM 154	CC0B	BR	GO SET CPU MODE, COND CODE 0
170C	8954	DYPE 203	TNRCNT	DPTT 013	UCENT	BR	GO SET UNIT CHECK, STORE CSW
1710	895E	DYPE 204	ZA1S1	DPTT 023	RESTES	BR	GO TEST UNIT STATUS
1712	3623	DYPE 205	ZA0S1			DO=DO\$K02	SET DO 6 FOR PTR RTN TO BR ON
1714	CE0F	DYPE 206		208	ZATEN	BR IF H00=1	BR IF ATTENTION BIT IS ON
1716	8924	DYPE 207		DPTT 049	SHARE	BR	SHARE PTR RTN, THEN STORE CSW
170E	8938	DYPE 208	ZATEN	DPTT 061	DEVE	BR	GO RESET I.B., THEN STORE CSW
		DYPE 209	*	HALT 10			
0EAC	A078	DYPE 210	HLTTW	ERRQ 083	RDCHST	BAL	GO READ CHANNEL AND UNIT STATUS
0EAE	2404	DYPE 211				SET MODE K=20	SET PR-KB MODE
0EBO	C423	DYPE 212		221	ZHACTV	BR IF G00=1	BR IF SECONDARY BIT IS SET

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0EB2	D139	DYPE 213		217	ZHTERM	BR IF G15=1	BR IF ACTIVE BIT IS SET
0EB4	EEA0	DYPE 214		216	ZHAO	BR IF TT2=0	BR IF NO INTERVENTION REQUIRED
0EB6	AAF4	DYPE 215		CCOM 165	CC3B	BR	GO SET CPU MODE, COND CODE 3
0EA0	88AC	DYPE 216	ZHAO	DCLA 226	STORO	BR	GO ZERO STATUS, STORE CSW
0EB8	1F00	DYPE 217	ZHTERM			RST TA K=80	RESET READ LATCH
0EBA	2F08	DYPE 218				SET TA K=40	SET WRITE LATCH
0EBC	3483	DYPE 219				GO=GO\$K08	SET O-COUNT BIT
0EBE	88D0	DYPE 220		DCLA 220	HOHO	BR	GO RST CC, CD FLAGS, STORE CSW
0EA2	D52A	DYPE 221	ZHACTV	225	ZHERE	BR IF G11=0	BR IF NO CC FLAG
0EA4	3E83	DYPE 222				HO=HO\$K08	SET CHANNEL END
0EA6	6664	DYPE 223				D=D+2	SET UCW ADDRESS TO UNIT STATUS
0EA8	7E6A	DYPE 224				STH H AS, D-2	STORE UNIT STATUS IN UCW
0EAA	88CE	DYPE 225	ZHERE	DCLA 219	DYPADD	BR	GO TEST CC FLAG
03A0	8DC4	DYPE 226	PRGCHK	CCOM 058	LSAVEB	BAL	GO SAVE P, I
03A2	A568	DYPE 227		DCOM 049	REDSTR	BAL	GO STORE UCW
03A4	9028	DYPE 228		168	GETUCW	BAL	GET UCW, SENSE BYTE, SET MODE
03A6	9F20	DYPE 229		010	NOATEN	BR	GO TO ENDING ROUTINE
		DYPE 230	*		NEW	TRANSLATE ROUTINE	
317A	5AFF	DYPE 231	STTRSL			H1=TI	GET KEYBOARD CHARACTER
317C	5F19	DYPE 232				U1=H1	MOVE CHAR TO WORK REG
317E	11C5	DYPE 233				U1=U1*-KCO	RESET CASE BITS -- 0,1
3180	CF10	DYPE 234		242	LOWCSE	BR IF H10=0	BR IF LOWER CASE
3182	2055	DYPE 235	BACK			U0=O\$K50	SET MODLE ADDR OF TABLE 50XX
3184	218D	DYPE 236				U1=U1+KBO	ADD XXB8
3186	218B	DYPE 237				U1=U1+K08	TO BIAS THE TABLE ADDRESS
3188	5F00	DYPE 238				RDB H1 AS,U	GET EBCDIC CHAR FROM TABLE
318A	4FFF	DYPE 239	LOWEQ			TE=H1	SEND CHAR TO 1052 DATA REG
318C	0F20	DYPE 240				RST TA K=02	SHARE RESET TO ALLOW 1052 CYCLE
318E	128E	DYPE 241				RTN	RETURN TO LINK ADDRESS
3190	C483	DYPE 242	LOWCSE	235	BACK	BR IF Z=0	BR IF CHAR IS A BLANK
3192	FC98	DYPE 243		246	LOWAQ	BR IF LZNZ	BR IF NOT DASH, AT, AMPERSAND
3194	11CB	DYPE 244				U1=U1KOC	MODIFY TABLE ADDR
3196	C482	DYPE 245		235	BACK	BR IF ZNZ	UNCONDITIONAL BRANCH
3198	E0B1	DYPE 246	LOWAQ	259	LOWFQ	BR IF HZ=0	BR IF NUMERIC OR POUND SIGN
319A	C820	DYPE 247		250	LOWBQ	BR IF H14=0	BR IF NOT KB 8 BIT
319C	DB03	DYPE 248		235	BACK	BR IF H15=1	BR IF KB 4 BIT
319E	EB2D	DYPE 249		256	LOWCQ	BR IF H16=1	BR IF KB 2 BIT
31A0	011F	DYPE 250	LOWBQ			Z=U1K11	
31A2	C4A9	DYPE 251		254	LOWDQ	BR IF Z=0	BR IF SLASH
31A4	1FB0	DYPE 252				H1=H1KBO	TRANSLATE ALPHA CHAR TO EBCDIC
31A6	C48A	DYPE 253		239	LOWEQ	BR IF ZNZ	
31A8	1F7D	DYPE 254	LOWDQ			H1=H1K70	TRANSLATE CHAR TO EBCDIC SLASH
31AA	C48A	DYPE 255		239	LOWEQ	BR IF ZNZ	
31AC	111B	DYPE 256	LOWCQ			U1=U1K01	MODIFY TABLE ADDR FOR COMMA,
		DYPE 257	*				DOLLAR SIGN, OR PERIOD
31AE	C482	DYPE 258		235	BACK	BR IF ZNZ	
31B0	CB34	DYPE 259	LOWFQ	261	LOWGQ	BR IF H14=0	BR IF NOT KB 8 BIT
31B2	EB39	DYPE 260		263	LOWHQ	BR IF H16=1	BR IF KB 2 BIT
31B4	1FFD	DYPE 261	LOWGQ			H1=H1KFO	TRANSLATE NUMERIC TO EBCDIC
31B6	C48A	DYPE 262		239	LOWEQ	BR IF ZNZ	MOD TABLE ADDR FOR O OR POUND
31B8	114B	DYPE 263	LOWHQ			U1=U1K04	
31BA	C482	DYPE 264		235	BACK	BR IF ZNZ	
		DYPE 265					

ADDR WORD SEQUENCE NO. LABEL NEXTSEQ NEXTLABEL STATEMENT

DYPE 266 * FOLLOWING IS THE 1052 XLATE TABLE
 DYPE 267 ATABLE ADDR=50B8
 50B8 4C7E DYPE 268 C XAUX'407E4C5E7A6C7D6E5C4D5D7F0000F07B4A6FE2E3E4E5'
 50CE E6E7 DYPE 269 C XAUX'E6E7E8E96B4F7C2500C06DD1D2D3D4D5D6D7D8D95B5A'
 50E4 6015 DYPE 270 C XAUX'601516004EC1C2C3C4C5C6C7C8C94B5F5005'
 DYPE 271 AEND

 * CROSS REFERENCE FOR CSECT DYPE *

DYPE 002 BSWI 062
 DYPE 008 DYPE 030
 DYPE 010 BWRP 089 DYPE 007 DYPE 025 DYPE 028 DYPE 033 DYPE 065 DYPE 229
 DYPE 015 DYPE 013
 DYPE 019 DYPE 017 DYPE 031
 DYPE 020 DYPE 002
 DYPE 021 DYPE 0C3
 DYPE 022 DYPE 0C6
 DYPE 025 DYPE 023
 DYPE 034 DYPE 022
 DYPE 039 DYPE 057
 DYPE 040 DYPE 037
 DYPE 047 DYPE 1C1
 DYPE 051 DYPE 048 DYPE 158 DYPE 162
 DYPE 053 DYPE 043 DYPE 111
 DYPE 054 DYPE 1C9
 DYPE 057 DYPE 054
 DYPE 062 DYPE 034 DYPE 059
 DYPE 064 DYPE 145
 DYPE 066 DYPE 041
 DYPE 067 DYPE 049 DYPE 062
 DYPE 071 DYPE 039 DYPE 110
 DYPE 083 DYPE 080
 DYPE 086 DYPE 083
 DYPE 089 DYPE 066
 DYPE 093 DYPE 1C5
 DYPE 094 DYPE 052
 DYPE 101 DYPE 059
 DYPE 102 DYPE 091
 DYPE 108 DYPE 1C3
 DYPE 110 DYPE 1C8
 DYPE 111 DYPE 054
 DYPE 113 DCLB 214
 DYPE 125 DYPE 123
 DYPE 129 DYPE 127
 DYPE 142 DYPE 131
 DYPE 144 DYPE 152
 DYPE 146 DYPE 144
 DYPE 150 DYPE 148
 DYPE 151 DYPE 133
 DYPE 154 DYPE 070
 DYPE 156 DYPE 138
 DYPE 157 DYPE 136
 DYPE 159 DYPE 140

* CROSS REFERENCE FOR CSECT DYPE *

DYPE 162	DYPE 160				
DYPE 163	DYPE 130				
DYPE 165	ERDR 104				
DYPE 168	DYPE 005	DYPE 117	DYPE 228		
DYPE 180	DYPE 015	DYPE 068	DYPE 086	DYPE 147	
DYPE 181	DYPE 051	DYPE 120	DYPE 164		
DYPE 184	DYPE 116				
DYPE 191	DCLA 308				
DYPE 203	DYPE 201				
DYPE 204	DYPE 199				
DYPE 205	DYPE 200				
DYPE 208	DYPE 206				
DYPE 210	DCLA 218				
DYPE 216	DYPE 214				
DYPE 217	DYPE 213				
DYPE 221	DYPE 212				
DYPE 225	DYPE 221				
DYPE 226	DCLR 037				
DYPE 231	ALDP 245	DYPE 096			
DYPE 235	DYPE 242	DYPE 245	DYPE 248	DYPE 258	DYPE 264
DYPE 239	DYPE 253	DYPE 255	DYPE 262		
DYPE 242	DYPE 234				
DYPE 246	DYPE 243				
DYPE 250	DYPE 247				
DYPE 254	DYPE 251				
DYPE 256	DYPE 249				
DYPE 259	DYPE 246				
DYPE 261	DYPE 259				
DYPE 263	DYPE 260				

ECOL DESCRIPTIVE TEXT

ENTRY POINTS

<p>MASK ENTRY IS MADE TO THIS LABEL FROM THE EPCH ROUTINE WHEN A COLUMN BINARY OPERATION NEEDS A MASK TO START. THE PUNCH ROW IMAGE BUFFER HAS BEEN FILLED WITH ONES, THE DATA ADDRESS AND COUNT HAVE BEEN STORED, 2540 MODE AND CPU ZONE HAVE BEEN SET.</p> <p>WLR ENTRY IS MADE TO THIS LABEL FROM THE EPXF ROUTINE WHEN A WRONG LENGTH RECORD HAS BEEN DETECTED. THE WLR BIT IS SET AND STORED IN THE INDICATOR BYTE IN AUXILIARY STORAGE. THE RESIDUAL COUNT IS STORED AND THE ERRQ ROUTINE IS BRANCHED TO FOR A UCW READOUT.</p> <p>MOVWLR ENTRY IS MADE FROM THE EPXF ROUTINE AFTER THE PUNCH DATA HAS BEEN TRANSLATED TO ROW IMAGE. THE RESIDUAL COUNT IS STORED AND THE ERRQ ROUTINE IS BRANCHED TO FOR A UCW READOUT.</p>	<p>GOCHAN ENTRY IS MADE FROM THE ERCX ROUTINE AFTER THE CCW COUNT IS ZERO FOR A DATA CHAINING OPERATION. A BRANCH TO DCLB IS MADE TO PICK UP THE NEXT CCW.</p> <p>CNTLRG ENTRY IS FROM THE EPCH ROUTINE DURING A DATA CHAINING OPERATION TO DETERMINE IF THE LAST CCW RESULTED IN AN 8 COLUMN GROUP BEING SET INTO THE PUNCH IMAGE BUFFER. IF AN 8 COLUMN GROUP WAS NOT COMPLETED BY THE LAST CCW, THE INFORMATION IS READ BACK FROM THE BUFFER, COMPLETED, AND STORED BACK INTO THE BUFFER.</p> <p>NEWMSK N ENTRY IS MADE TO THIS BRANCH SET FROM THE EPXF ROUTINE TO FETCH A NEW COLUMN MASK AS INDICATED BY THE CONTROL REGISTER -10-.</p>
---	--

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ECOL 001	T			PUNCH COLUMN BINARY AND DATA CHAINING	
		ECOL 002	*				
194A	24E5	ECOL 003	MASK			GO=O\$K80	FIRST MASK
194C	2AF7	ECOL 004				TO=O\$KFF	RST ALL REG'S
194E	5AB9	ECOL 005				T1=TO	
1950	4EA6	ECOL 006				H=T	
1952	5B99	ECOL 007				I1=T1	
1954	5B59	ECOL 008				G1=T1	
1956	01E3	ECOL 009	AGAIN			Z=U1*-KOE	MASK FOR ODD CT
1958	FOE1	ECOL 010		014	CTEVEN	BR IF LZ=0	BR IF CCW CT EVEN
195A	211B	ECOL 011				U1=U1+K01	MAKE CT EVEN
195C	D861	ECOL 012		014	CTEVEN	BR IF P0 BIT5=1	BR IF TOTAL CT EVEN
195E	311B	ECOL 013				U1=U1-K01 **TOTAL CT	ODD, TAKE 1 COL LESS
1960	E824	ECOL 014	CTEVEN	022	RDDATA	BR IF P0 BIT6=0	BR IF LOWER HALF
1962	261B	ECOL 015				DO=DO+K01	WLR CTR + 1
1964	F4A4	ECOL 016		022	RDDATA	BR IF AC=0	BR IF NOT WLR
1966	2DA1	ECOL 017	WLR			P1=O-K0A	WLR, P=10F5
1968	5EC0	ECOL 018				RDH H AS, P+0	RD IND BYTE
196A	3E13	ECOL 019				HO=HO\$K01	WLR INDICATOR
196C	7EC0	ECOL 020				STH H AS, P+0	
196E	997C	ECOL 021		059	HALF	BR	
1924	5738	ECOL 022	RDDATA			RDB D1 V+1	READ DATA

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1926	E72A	ECOL 023		025	N	BR IF D1 BIT2=0	
1928	6541	ECOL 024				G1=G1□GO	
192A	F72E	ECOL 025	N	027	NE	BR IF D1 BIT3=0	
192C	6B41	ECOL 026				T1=T1□GO	
192E	C332	ECOL 027	NE	029	NEX	BR IF D1 BIT4=0	
1930	6F41	ECOL 028				H1=H1□GO	
1932	D336	ECOL 029	NEX	031	NEXT	BR IF D1 BIT5=0	
1934	6941	ECOL 030				I1=I1□GO	
1936	E33A	ECOL 031	NEXT	033	NEXTB	BR IF D1 BIT6=0	
1938	6A41	ECOL 032				T0=T0□GO	
193A	F33E	ECOL 033	NEXTB	035	NEXTBT	BR IF D1 BIT7=0	
193C	6E41	ECOL 034				H0=H0□GO	
193E	5224	ECOL 035	NEXTBT			V=V+1	EVERY OTHER BYTE
1940	282B	ECOL 036				I0=I0+K02	CONTROL REG + 2
1942	311B	ECOL 037				U1=U1-K01	CCW COUNT - 2
1944	C48D	ECOL 038		041	STORE	BR IF Z=0	BR IF CT=0
1946	5879	ECOL 039	TEMPRY			D1=I0	
1948	B345	ECOL 040		124	NEWMSK N	N=D1L	BREAK FOR NEW MASK
190C	75C0	ECOL 041	STORE			STB G1 AS,P+0	STORE INTO PCH IMAGE
190E	2DAB	ECOL 042				P1=P1+K0A	BUFFER
1910	7BC0	ECOL 043				STB T1 AS,P+0	
1912	2DAB	ECOL 044				P1=P1+K0A	
1914	7FC0	ECOL 045				STB H1 AS,P+0	
1916	1C1B	ECOL 046				P0=P0□K01	**INVERT 1ST 3 OR 2ND 3 IND
1918	F871	ECOL 047		053	DON2ND	BR IF P0 BIT7=1	BR IF STORE 2ND 3
191A	5959	ECOL 048				G1=I1	MOVE 2ND 3 BYTES
191C	5AB9	ECOL 049				T1=T0	
191E	5EF9	ECOL 050				H1=H0	
1920	2DAB	ECOL 051				P1=P1+K0A	
1922	990C	ECOL 052		041	STORE	BR	GO TO STORE 2ND 3
1970	2DCD	ECOL 053	DON2ND			P1=P1+KCO	
1972	2DFB	ECOL 054				P1=P1+KOF	POINT TO NEXT 6 COL'S
1974	5119	ECOL 055				U1=U1	
1976	C4FD	ECOL 056		059	HALF	BR IF Z=0	BR IF CCW CT=0
1978	08AD	ECOL 057				Z=I0□KAO	FORCE LOWER TO STOP
197A	E0CA	ECOL 058		003	MASK	BR IF HZNZ	IN CASE OF WLR
197C	1C2B	ECOL 059	HALF			P0=P0□K02	INVERT UPPER OR LOWER HALF BIT
197E	E814	ECOL 060		092	LOHALF	BR IF P0 BIT6=0	BR IF LOWER HALF
1980	5629	ECOL 061	MOVMLR			V0=V0	MOVE WLR CTR
1982	71A2	ECOL 062				STB U1 DA,AC	STORE RESIDUE COUNT
1984	5C99	ECOL 063				I1=P0	STORE 1X REG
1986	A062	ECOL 064		ERRQ 087	RPHUCW	BAL	READ PUNCH UCW
1988	A044	ECOL 065		ERRQ 092	RDIND	BAL	READ INDICATOR AND SENE
198A	2EC7	ECOL 066				H0=0	ZERO UNIT STATUS
198C	C538	ECOL 067		072	NOCD	BR IF G1 BIT0=0	BR IF NOT CDA
198E	F837	ECOL 068		071	OVERCT	BR IF P0 BIT7=1	BR IF WLR - OVER COUNT
1990	52D9	ECOL 069	GOCHAN			P1=V0	CDA, STORE WLR CTR IN P1
1992	93CA	ECOL 070		DCLB 039	CHAIN	BR	GO CHAIN
1986	1525	ECOL 071	OVERCT			G1=G1*-K20	SUPPRESS SLI
1988	3485	ECOL 072	NOCD			G0=G0\$K80	SECONDARY BIT
198A	5FA0	ECOL 080	STKOLD			RDB H1 AS,T+0	GET OLD STACKER INFO
198C	7CA8	ECOL 081	STKNEW			STH P AS,T+2	STORE NEW STACKER INFO
198E	9F01	ECOL 082		083	STKSEL N	N=H1 BITS123	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1800	2FC8	ECOL 083	STKSEL 0			SET P K=40	POCKET P1(RU031)
1802	98CA	ECOL 084		088	COUNT	BR	
1804	2FOC	ECOL 085	STKSEL 2			SET P K=60	POCKET P2(RU31&025)
1806	980A	ECOL 086		088	COUNT	BR	
1808	2FOA	ECOL 087	STKSEL 4			SET P K=50	POCKET RP3(RU31&025)
180A	221B	ECOL 088	COUNT			VO=VO+K01	INCREMENT WLR CTR
180C	F491	ECOL 089		091	NOUNCT	BR IF NO UNDER COUNT	
180E	3C13	ECOL 090				PO=PO\$K01	UNDER COUNT, SET WLR INDICATOR
1810	911A	ECOL 091	NOUNCT	EXFR 122	WLRBR	BR	INTERAGATE CHAINING AND SLI
1994	5C89	ECOL 092	LOHALF			IO=UO	RESTORE CONTROL CTR
1996	51A2	ECOL 093				RDB U1 DA, AC	RESTORE CCW ADJUSTED COUNT
1998	52E2	ECOL 094				RDH V DA, BC	RESTORE DATA ADDR
199A	D81E	ECOL 095		097	OKCT	BR IF PO BIT5=0	BR IF TOTAL CT ODD
199C	5224	ECOL 096				V=V+1	TOTAL CT EVEN
199E	01E3	ECOL 097	OKCT			Z=U1*-K0E	MASK FOR ODD COUNT
19A0	FOAB	ECOL 098		103	EVENCT	BR IF LZ=0	BR IF EVEN CT
19A2	21FF	ECOL 099				U1=U1+KFF	MAKE IT EVEN
19A4	D829	ECOL 100		102	ODDNOT	BR IF PO BIT5=1	BR IF TOTAL CT EVEN
19A6	212B	ECOL 101				U1=U1+K02	**TOTAL CT ODD, MAKE 1 COL MORE
19A8	1C4B	ECOL 102	ODDNOT			PO=PO+K04	**ODD CT, FLIP TOTAL ODD CT BIT
19AA	58D5	ECOL 103	EVENCT			P1=IOXL	**RECOVER PCT IMAGE BUFFER ADDR
19AC	2DCB	ECOL 104				P1=P1+K0C	
19AE	2D3D	ECOL 105				P1=P1+K30	
19B0	5889	ECOL 106	CNTRG			IO=IO	
19B2	FOC0	ECOL 107		109	COMDLE	BR IF LZNZ	
19B4	994A	ECOL 108		003	MASK	BR	
19C0	2D3D	ECOL 109	COMDLE			P1=P1+K30	**CD STARTED FROM THE MIDDLE OF
19C2	2D2B	ECOL 110				P1=P1+K02	**ROW FORM BYTES, CHANGE ADDR
19C4	5FC0	ECOL 111	READ			RDB H1 AS, P+0	TO READ THEM OUT BACKWARDS
19C6	3D9B	ECOL 112				P1=P1-K09	
19C8	5BC0	ECOL 113				RDB T1 AS, P+0	
19CA	3D9B	ECOL 114				P1=P1-K09	
19CC	55C0	ECOL 115				RDB G1 AS, P+0	
19CE	1C1B	ECOL 116				PO=PO+K01	**INVERT 1ST 3 OR 2ND 3 IND
19D0	F854	ECOL 117		119	SECOND	BR IF PO BIT7=0	BR IF RD 1ST 3 BYTES
19D2	9546	ECOL 118		039	TEMPRY	BR	RD ALL 6 BYTES, GO
19D4	5FE9	ECOL 119	SECOND			HO=H1	MOVE THE LOWER 3 BYTE
19D6	5BA9	ECOL 120				TO=T1	
19D8	5599	ECOL 121				I1=G1	
19DA	3D9B	ECOL 122				P1=P1-K09	SUBT 0A ONCE MORE
19DC	99C4	ECOL 123		111	READ	BR	**GO READ THE UPPER 3 BYTES
1A40	C862	ECOL 124	NEWMSK 0	141	NOTPOL	BR IF PO BIT4=0	BR IF FROM EPXF
1A42	990C	ECOL 125		041	STORE	BR	**DONE 8 COLUMN, GO STORE THEM
1A44	2445	ECOL 126	NEWMSK 2			GO=0\$K40	
1A46	9A5E	ECOL 127		139	BRNCH	BR	
1A48	2425	ECOL 128	NEWMSK 4			GO=0\$K20	
1A4A	9A5E	ECOL 129		139	BRNCH	BR	
1A4C	2415	ECOL 130	NEWMSK 6			GO=0\$K10	
1A4E	9A5E	ECOL 131		139	BRNCH	BR	
1A50	2483	ECOL 132	NEWMSK 8			GO=0\$K08	
1A52	9A5E	ECOL 133		139	BRNCH	BR	
1A54	2443	ECOL 134	NEWMSK A			GO=0\$K04	
1A56	9A5E	ECOL 135		139	BRNCH	BR	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1A58	2423	ECOL 136	NEWMSK C			GO=0\$K02	
1A5A	5A5E	ECOL 137		139	BRNCH	BR	
1A5C	2413	ECOL 138	NEWMSK E			GO=0\$K01	
1A5E	C864	ECOL 139	BRNCH	142	POLNOT	BR IF P0 BIT4=0	BR IF FROM EPXF
1A60	5956	ECOL 140		009	AGAIN	BR	
1A62	9B30	ECOL 141	NOTPOL	EPXF 069	STORE	BR	**GOT NEW MASK BACK TO PXFR
1A64	5B66	ECOL 142	POLNOT	EPXF 016	UPDATE	BR	**GOT NEW MASK BACK TO PXFR

 * CROSS REFERENCE FOR CSECT ECOL *

ECOL 003	ECOL 058	ECOL 108	EPCH 103				
ECOL 009	ECOL 140						
ECOL 014	ECOL 010	ECOL 012					
ECOL 017	EPXF 019						
ECOL 022	ECOL 014	ECOL 016					
ECOL 025	ECOL 023						
ECOL 027	ECOL 025						
ECOL 029	ECOL 027						
ECOL 031	ECOL 029						
ECOL 033	ECOL 031						
ECOL 035	ECOL 033						
ECOL 039	ECOL 118						
ECOL 041	ECOL 038	ECOL 052	ECOL 125				
ECOL 053	ECOL 047						
ECOL 059	ECOL 021	ECOL 056					
ECOL 061	EPXF 082						
ECOL 069	ERCX 039						
ECOL 071	ECOL 068						
ECOL 072	ECOL 067						
ECOL 083	ECOL 082						
ECOL 088	ECOL 084	ECOL 086					
ECOL 091	ECOL 089						
ECOL 092	ECOL 060						
ECOL 097	ECOL 095						
ECOL 102	ECOL 100						
ECOL 103	ECOL 098						
ECOL 106	EPCH 086						
ECOL 109	ECOL 107						
ECOL 111	ECOL 123						
ECOL 119	ECOL 117						
ECOL 124	ECOL 040	EPXF 040					
ECOL 139	ECOL 127	ECOL 129	ECOL 131	ECOL 133	ECOL 135	ECOL 137	
ECOL 141	ECOL 124						
ECOL 142	ECOL 139						

EPCH DESCRIPTIVE TEXT

ENTRY POINTS

NPCH ENTRY IS MADE HERE FROM THE ERDR ROUTINE DURING A PUNCH WRITE START I/O OR A PFR. THE INDICATOR AND SENSE BYTES ARE SET UP AND SAVED IN LOCAL STORAGE.

HIGHCT ENTRY IS MADE FROM THE ERDR ROUTINE WITH A NEW CCW BYTE COUNT, TO DETERMINE IF AN EXCESS COUNT HAS BEEN GIVEN. IF THERE IS AN EXCESS COUNT, THE EXCESS IS STORED IN AUX OXBE.

PICKUP ENTRY IS MADE FROM THE ERDR ROUTINE TO TEST FOR UNUSUAL COMMAND SEQUENCE. THE READ INDICATOR AND WLR COUNTER ARE SET, ANY EXCESS BYTE COUNT IS STORED IN AUX OXBE.

NATIVE 2540 PUNCH UCW FORMAT AND AUX LOCATIONS

INDICATORS AND SENSE BYTE AND AUX LOCATIONS

00E8	00E9	00EA	00EC-ED	00EE-FF
* CHANNEL * * STATUS *	* CHANNEL * * FLAGS *	* UNIT * * STATUS *	* NOT * * USED *	* RESIDUAL * * COUNT *

10F4	10F5	10F2
* INDICATOR *	* SENSE BYTE *	* OLD * * STACKER *

CHANNEL STATUS	CHANNEL FLAGS	UNIT STATUS
0=SECONDARY 1=WLR 2=PRG CHK 3=PKOT CHK 4=NOT USED 5=NOT USED 6=INTFCE CHK 7=INT IN BFR	0=CHAIN DATA 1=CHAIN CMD 2=SLI 3=SKIP 4=PCI 5=ACTIVE 6=NOT USED 7=NOT USED	0=NOT USED 1=NOT USED 2=NOT USED 3=BUSY 4=CH END 5=DEV END 6=UNIT CHK 7=UNIT EXCEPT

INDICATOR	SENSE BYTE	OLD STACKER
0=CHAIN DATA 1=STK SEL 2=STK SEL 3=0 4=PFR READ 5=COUNT STOR 6=PFR COUNT 7=WLR	0=CMD REJ 1=NOT RDY 2=NOT USED 3=EQUIP CHK 4=VALIDITY CHK 5=NOT USED 6=UNUSUAL CMD 7=NOT USED	0=NOT USED 1=STK SEL 2=STK SEL 3=0 4=NOT USED 5=NOT USED 6=NOT USED 7=NOT USED

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		EPCH 001	T			PUNCH SID CMD DECODE & CHAINING	
		EPCH 002	*			FOLLOWING INFORMATION--	
		EPCH 003	*				
		EPCH 004	*				
		EPCH 005	*			U REG CONTAINS - COUNT	
		EPCH 006	*			V REG CONTAINS - DATA ADDRESS	
		EPCH 007	*			G0 REG CONTAINS - CHANNEL STATUS	
		EPCH 008	*			G1 REG CONTAINS - FLAG	
						H0 REG CONTAINS - COMMAND BYTE	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		EPCH 009	*				{H0=00 IF DATA CHAINING}
12D4	1F00	EPCH 010	NPCH			RST P K=80	RST PCH & VAL CHKS(RU031)
12D6	FA74	EPCH 014		043	READ	BR IF H0 BIT7=0	BR TO CHECK IF READ COMMAND
12D8	6CF7	EPCH 016				PU=PO*H1	CLEAR STKR
12DA	EA53	EPCH 017		032	COMREJ	BR IF H0 BIT6=1	BR IF 7,6 COMBINATION
12DC	CE62	EPCH 018		021	STK1BT	BR IF H0 BIT0=0	BR IF STKR P1 OR P2
12DE	DE53	EPCH 019		032	COMREJ	BR IF H0 BIT1=1	BR IF 0&1 BITS IN COMMAND
12E0	3C45	EPCH 020				PO=PO\$K40	RP3
12E2	DE66	EPCH 021	STK1BT	023	SSDONE	BR IF H0 BIT1=0	
12E4	3C25	EPCH 022				PO=PO\$K20	P2
12E6	E853	EPCH 023	SSDONE	032	COMREJ	BR IF PO BIT6=1	BR IF PREVIOUS PFR WRITE
12E8	CA48	EPCH 024		034	NOTPFR	BR IF H0 BIT4=0	BR IF THIS CMD NOT PFR
12EA	3C23	EPCH 026				PO=PO\$K02	PFR WRITE CMD,SET INDICATOR
12EC	2F40	EPCH 027				SET P K=04	SET PFR RESTART(RU025)
12EE	2462	EPCH 028				SET MODE K=16	PUNCH MODE AND ZONE
12FO	1643	EPCH 029				DO=DO*-K04	PFR INDICATOR FOR PCH TRAP(ETRP)
12F2	92CC	EPCH 030		036	MODE	BR	
12D2	98B4	EPCH 032	COMREJ	ERDR 138	CMDREJ	BR	
12C8	2462	EPCH 034	NOTPFR			SET MODE K=16	PCHMODE AND ZONE
12CA	3643	EPCH 035				DO=DO\$K04	SET NOT PFR INDICATOR
12CC	2402	EPCH 036	MODE			SET MODE K=10	
12CE	1C83	EPCH 037				PO=PO*-K08	SET FEED INDICATOR(RST PFR READ)
12D0	930C	EPCH 041		056	SETCTR	BR	
12F4	EA52	EPCH 043	READ	032	COMREJ	BR IF H0 BIT6=0	
12F6	DA53	EPCH 044		032	COMREJ	BR IF H0 BIT5=1	BR IF 5-6 BIT COMBINATION
12F8	0E49	EPCH 045				Z=H0+K40	
12FA	F4D2	EPCH 046		032	COMREJ	BR IF AC=0	BR IF 0&1 BITS NOT ON
12FC	EE53	EPCH 047		032	COMREJ	BR IF H0 BIT2=1	BR IF COL BIN
12FE	EF84	EPCH 048		052	PICKUP	BR IF PS BIT2=0	BR IF NOT UNIT EXCEPTION
1300	2C07	EPCH 049				PO=0	PFR UNIT EXCEPTION,ALLOW FEED
1302	98B0	EPCH 050		ERDR 133	UEXPFR	BR	SHARE UNIT EXCEPTION READER
1304	C808	EPCH 052	PICKUP	054	SETP4	BR IF PO BIT4=0	BR IF PREVIOUS CMD IS A FEED
1306	2D23	EPCH 053				P1=0\$K02	PREV A READ,SET UNUSUAL CMD BIT
1308	3C83	EPCH 054	SETP4			PO=PO\$K08	SET READ INDICATOR
130A	1C23	EPCH 055				PO=PO*-K02	RST FEED OR PFR WRITE INDICATOR
130C	3659	EPCH 056	SETCTR			DO=0-K50	SET WLR DO=AF
130E	5009	EPCH 057	HIGHCT			U0=U0	
1310	C4A3	EPCH 058		067	HICT0	BR IF Z=0	BR IF NO EXCESS COUNT
1312	3C53	EPCH 059				PO=PO\$K05	EXCESS COUNT & WLR INDICATORS
1314	4A06	EPCH 060				T=U	MOVE COUNT TO T REG
1316	2007	EPCH 061				U0=0	GET
1318	21A5	EPCH 062				U1=0\$KA0	EXCESS COUNT
131A	2002	EPCH 063				SET S3	
131C	7B19	EPCH 064				TIC=T1-U1+C	AND
131E	7A09	EPCH 065				TOC=TO-U0+C	STORE
1320	7AF2	EPCH 066				STH T DA, BE	IT
1322	8564	EPCH 067	HICT0	ERRQ 102	STIND	BAL	STORE
1324	71A2	EPCH 068				STB U1 DA, AC	STORE ADJUSTED COUNT
1326	72E2	EPCH 069				STH V DA, BC	STORE DATA ADDRESS
1328	F72C	EPCH 070		073	STAY	BR IF D1 BIT3=0	BR IF PUNCH
132A	98C4	EPCH 071	GOBACK	ERDR 146	BACK	BR	READER,GO TO ERDR ROUTINE
132C	C82B	EPCH 073	STAY	071	GOBACK	BR IF PO BIT4=1	BR IF PFR READ(ICC OR SID)
132E	CC42	EPCH 074		087	CCSID	BR IF PO BIT0=0	BR IF NOT CDA

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1330	5CD9	EPCH 078				P1=PO	MOVE IND TO P1 TO SAVE CDA IND
1332	59C9	EPCH 079				PO=I1	CDA,RESTORE PO
1334	C85E	EPCH 080		101	GOEPXF	BR IF PO BIT4=0	CDA FOR NO COL
1336	58C9	EPCH 081				UO=IO	STORE CONTROL REG
1338	D83F	EPCH 082		085	TOTEVN	BR IF PO BIT5=1	BR IF TOTAL COUNT EVEN
133A	5224	EPCH 083				V=V+1	LAST TOTAL COUNT
133C	282B	EPCH 084				IO=IO+K02	WAS ODD
133E	58D5	EPCH 085	TOTEVN			P1=IOXL	RECOVER BUFFER ADDRESS
1340	99B0	EPCH 086		ECOL 106	CNTLRG	BR	
1342	28C7	EPCH 087	CCSIO			T1=0	CC OR SIO, T=1000
1344	2C45	EPCH 088				PO=0\$K40	
1346	2DC3	EPCH 089				P1=0\$K0C	P=400C
1348	28F7	EPCH 090				IO=0\$KFF	
134A	5899	EPCH 091				I1=IO	I=FFFF
134C	54A0	EPCH 092	LOOP			RDH G AS,T+0	XFER PCH IMAGE AREA(1000) TO
134E	78A8	EPCH 093				STH I AS,T+2	PCH CHK AREA (400C) AND STORE
1350	74C8	EPCH 094				STH G AS,P+2	FF BACK INTO PCH IMAGE(1000)
1352	CD4C	EPCH 095		092	LOCP	BR IF P1 BIT0=0	BR IF NOT DONE
1354	D94C	EPCH 096		092	LOCP	BR IF P1 BIT5=0	BR IF NOT DONE
1356	5884	EPCH 097				I=I+1	DONE,I=0, P=4084
1358	3CE9	EPCH 098				PO=0\$KE0	
135A	2DC7	EPCH 099				P1=0	P=1F00
135C	EE63	EPCH 100		103	GOECOL	BR IF H0 BIT2=1	
135E	1CA3	EPCH 101	GOEPXF			PO=PO*-K0A	P=15XX, DATA MODE 1 WRITE
1360	9B54	EPCH 102		EPXF 003	PXFR	BR	SIC, CC OR CDA
1362	994A	EPCH 103	GOECOL	ECOL 003	MASK	BR	COL BIN,SIO,CC

 * CROSS REFERENCE FOR CSECT EPCH *

EPCH 010	ERDR 095						
EPCH 021	EPCH 018						
EPCH 023	EPCH 021						
EPCH 032	EPCH 017	EPCH 019	EPCH 023	EPCH 043	EPCH 044	EPCH 046	EPCH 047
EPCH 034	EPCH 024						
EPCH 036	EPCH 030						
EPCH 043	EPCH 014						
EPCH 052	EPCH 048	ERDR 130					
EPCH 054	EPCH 052						
EPCH 056	EPCH 041						
EPCH 057	ERDR 059						
EPCH 067	EPCH 058						
EPCH 071	EPCH 073						
EPCH 073	EPCH 070						
EPCH 085	EPCH 082						
EPCH 087	EPCH 074						
EPCH 092	EPCH 055	EPCH 096					
EPCH 101	EPCH 080						
EPCH 103	EPCH 100						

EPXF DESCRIPTIVE TEXT

ENTRY POINTS

<p>PXFR ENTRY IS MADE HERE FROM THE EPCH ROUTINE TO SET UP THE WORK AREA AND COUNTER. THE DATA TO BE PUNCHED IS TRANSFERED AND SET INTO THE ROW IMAGE BUFFER IN COMPLEMENT FORM. SEE FEMDM DIAGRAM 5-114.</p> <p>UPDATE ENTRY IS MADE HERE FROM THE ECOL ROUTINE TO INCREMENT THE WLR COUNTER AFTER A NEW COLUMN MASK HAS BEEN SET. IF THE WLR COUNTER OVERFLOWS, THE ECOL ROUTINE IS BRANCHED TO AND THE WLR BIT IS SET.</p>	<p>STORE ENTRY IS MADE HERE FROM THE ECOL ROUTINE TO STORE THE 12, 11, 0, 8, AND 9 ROW WORK REGISTERS INTO THE ROW IMAGE BUFFER. IF THE CCW COUNT IS NOW ZERO, A BRANCH IS TAKEN TO THE ECOL ROUTINE. IF THE COUNT IS NOT ZERO, A NEW COLUMN MASK IS SET UP AND THE BUILDING OF THE PUNCH ROW IMAGE CONTINUES.</p>
---	---

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		EPXF 001	T			PUNCH TRANSFER(EBCDIC TO ROW FORM) & CHAINING	
		EPXF 002	*				
1854	51C9	EPXF 003	PXFR			U0=U1	MOVE CCW CT
1856	2A65	EPXF 007				T0=0\$K60	X'LATE TABLE ADDR FOR NOT 24K
1858	CD6F	EPXF 009		024	CDA	BR IF P1 BIT0=1	BR IF DATA CHAIN
185A	21F7	EPXF 010	CLEAR			U1=0\$KFF	CLEAR
185C	5179	EPXF 011				D1=U1	ALL THE
185E	51F9	EPXF 012				H1=U1	ACCUMULATING
1860	51E9	EPXF 013				H0=U1	REGISTERS
1862	5199	EPXF 014				I1=U1	
1864	2485	EPXF 015				G0=0\$K80	FIRST MASK
1866	261B	EPXF 016	UPDATE			D0=D0+K01	INCREMENT WLR CTR
1868	F4CC	EPXF 017		020	READ	BR IF AC=0	
186A	5019	EPXF 018				U1=U0	
186C	9966	EPXF 019		ECOL 017	WLR	BR	
184C	5B38	EPXF 020	READ			RDB T1 V+1	READ FROM MAIN STORE
184E	55A0	EPXF 021				RDB G1 AS,T+0	XLATE TO CONDENSED
1850	157B	EPXF 022				G1=G1^K07	INVERTS BITS 4,2,1
1852	9115	EPXF 023		041	BIT421 N	N=G1 BITS567	
186E	58D5	EPXF 024	CDA			P1=IOXL **CHAINING, RECOVER BUFFER ADR	
1870	5889	EPXF 025				IO=IO	
1872	F0DB	EPXF 026		010	CLEAR	BR IF LZ=0 **BR IF LAST CHAIN ENDED AT 8	
		EPXF 027	*				BYTES BOUNDARY
1874	51C0	EPXF 028				RDB U1 AS,P+0	LAST CHAIN(OR SIO)
1876	2DAB	EPXF 029				P1=P1+K0A	ENDED IN THE MIDDLE
1878	57C0	EPXF 030				RDB D1 AS,P+0	OF A 8 BYTES BOUN-
187A	2DAB	EPXF 031				P1=P1+K0A	DARY, READ THE 12,
187C	59C0	EPXF 032				RDB I1 AS,P+0	11, 0, 8, & 9TH
187E	2D5D	EPXF 033				P1=P1+K50	BYTES OUT
1880	5FC0	EPXF 034				RDB H1 AS,P+0	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1882	2DAB	EPXF 035				P1=P1+K0A	
1884	55C0	EPXF 036				RDB G1 AS,P+0	
1886	55E9	EPXF 037				H0=G1	
1888	58D5	EPXF 038				P1=IOXL	RECOVER BUFFER ADDR
188A	5859	EPXF 039	TEMPRY			G1=IO	
188C	B145	EPXF 040		ECOL 124	NEWMASK N	N=G1L	GO GET NEW MASK
1A80	2DAB	EPXF 041	BIT421 0			P1=P1+K0A	ADD 60(90) - 7 PUNCH
1A82	2DAB	EPXF 042	BIT421 1			P1=P1+K0A	ADD 50(80) - 6 PUNCH
1A84	2DAB	EPXF 043	BIT421 2			P1=P1+K0A	ADD 46(70) - 5 PUNCH
1A86	2DAB	EPXF 044	BIT421 3			P1=P1+K0A	ADD 3C(60) - 4 PUNCH
1A88	2DAB	EPXF 045	BIT421 4			P1=P1+K0A	ADD 32(50) - 3 PUNCH
1A8A	2DAB	EPXF 046	BIT421 5			P1=P1+K0A	ADD 28(40) - 2 PUNCH
1A8C	9BC8	EPXF 047	BIT421 6	049	ADD1E	BR	GO ADD 1E(30)-1 PUNCH
1A8E	9B14	EPXF 048	BIT421 7	055	BYTES	BR	NO PUNCH IN 1 THRU 7
1B08	2D1D	EPXF 049	ADD1E			P1=P1+K10	ADD 1E TO BUFFER ADDR
1B0A	2DEB	EPXF 050				P1=P1+K0E	
1B0C	58C0	EPXF 051				RDB I1 AS,P+0	STORE
1B0E	6B41	EPXF 052				I1=I1GO	1 THRU 7
1B10	78C0	EPXF 053				STB I1 AS,P+0	PUNCH
1B12	58D5	EPXF 054				P1=IOXL	RECOVER PCH BFR ADDR
1B14	C518	EPXF 055	BYTES	057	N	BR IF G1 BIT0=0	
1B16	6141	EPXF 056				U1=U1GO	12TH PUNCH
1B18	D51C	EPXF 057	N	059	NE	BR IF G1 BIT1=0	
1B1A	6741	EPXF 058				D1=D1GO	11TH PUNCH
1B1C	E520	EPXF 059	NE	061	NEX	BR IF G1 BIT2=0	
1B1E	6941	EPXF 060				I1=I1GO	0TH PUNCH
1B20	F524	EPXF 061	NEX	063	NEXT	BR IF G1 BIT3=0	
1B22	6F41	EPXF 062				H1=H1GO	8TH PUNCH
1B24	C128	EPXF 063	NEXT	065	DONE	BR IF G1 BIT4=0	
1B26	6E41	EPXF 064				H0=H0GO	9TH PUNCH
1B28	282B	EPXF 065	DONE			IO=IO+K02	ADD 2 TO CONTROL CTR
1B2A	20FF	EPXF 066				U0=U0+KFF	DECREMENT CCW CT
1B2C	C4B1	EPXF 067		069	STORE	BR IF Z=0	
1B2E	5B8A	EPXF 068		039	TEMPRY	BR	NOT DONE,GET NEW MASK
1B30	71C0	EPXF 069	STORE			STB U1 AS,P+0	STORE 12TH BYTE
1B32	2DAB	EPXF 070				P1=P1+K0A	
1B34	77C0	EPXF 071				STB D1 AS,P+0	STORE 11TH BYTE
1B36	2DAB	EPXF 072				P1=P1+K0A	
1B38	79C0	EPXF 073				STB I1 AS,P+0	STORE 0TH BYTE
1B3A	2D5D	EPXF 074				P1=P1+K50	
1B3C	7FC0	EPXF 075				STB H1 AS,P+0	STORE 8TH BYTE
1B3E	2DAB	EPXF 076				P1=P1+K0A	
1B40	5EF9	EPXF 077				H1=H0	
1B42	7FC0	EPXF 078				STB H1 AS,P+0	STORE 9TH BYTE
1B44	58D5	EPXF 079				P1=IOXL	RESTORE BFR ADDR
1B46	5019	EPXF 080				U1=U0	
1B48	C4DA	EPXF 081		010	CLEAR	BR IF ZNZ	BR IF CTR NOT ZERO
1B4A	5980	EPXF 082		ECOL 061	MOVWLR	BR	FINISHED
		EPXF 083	*				NATIVE 2540 PUNCH TRANSLATE TABLE
		EPXF 085	ATABLE	ADDR=6000			
6000	B989	EPXF 089	C			X AUX* B9898A8B8C8D8E8F98999A9B9C9D9E9F9D9494A484C4D*	
6016	4E4F	EPXF 090	C			X AUX* 4E4F58595A5B5C5D5E5F79292A2B2C2D2E2F38393A3B*	
602C	3C3D	EPXF 091	C			X AUX* 3C3D3E3FF9090A0B0C0D0E0F18191A1B1C1D1E1F00A9*	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
6042	AAAB	EPXF 092	C			X AUX* AAABACADAEAFB89192939495969780C9CACBCCCDCECF*	
6058	D851	EPXF 093	C			X AUX* D85152535455565740216A686C6D6E6F7831C0333435*	
606E	3637	EPXF 094	C			X AUX* 3637E0E9EAEBECEDEEFF81112131415161781A1A2A3*	
6084	A4A5	EPXF 095	C			X AUX* A4A5A6A7B0A8B2B3B4B5B6B7D1C1C2C3C4C5C6C7DOC8*	
609A	D2D3	EPXF 096	C			X AUX* D2D3D4D5D6D771616263646566677068727374757677*	
60B0	F1E1	EPXF 097	C			X AUX* F1E1E2E3E4E5E6E7F0E8F2F3F4F5F6F7A08182838485*	
60C6	8687	EPXF 098	C			X AUX* 86879088BABBBCBDBEBF60414243444546475048DADB*	
60DC	DCDD	EPXF 099	C			X AUX* DCDDEDF326922232425262730287A7B7C7D7E7F2001*	
60F2	0203	EPXF 100	C			X AUX* 0203040506071008FAFBFCFDFEFF*	
		EPXF 101	AEND				

 * CROSS REFERENCE FOR CSECT EPXF *

EPXF 003	EPCH 102		
EPXF 010	EPXF 026	EPXF 081	
EPXF 016	ECOL 142		
EPXF 020	EPXF 017		
EPXF 024	EPXF 009		
EPXF 039	EPXF 068		
EPXF 041	EPXF 023		
EPXF 049	EPXF 047		
EPXF 055	EPXF 048		
EPXF 057	EPXF 055		
EPXF 059	EPXF 057		
EPXF 061	EPXF 059		
EPXF 063	EPXF 061		
EPXF 065	EPXF 063		
EPXF 069	ECOL 141	EPXF 067	

ERCX DESCRIPTIVE TEXT

ENTRY POINTS

START	ENTRY IS MADE FROM THE ERDR ROUTINE TO START READER COLUMN IMAGE TO STORAGE TRANSFER. SEE FEMDM DIAGRAM 5-105.	CDFLAG	ENTRY IS FROM THE ERDR ROUTINE WHEN THE SKIP OPERATION OF A READ OR PFR READ IS COMPLETED, (CCW COUNT=0). IF NOT DATA CHAINING, CONTINUE WITH THE NORMAL ENDING PROCEDURE. IF DATA CHAINING PREPARE TO FETCH A NEW CCW.
WLRSET	ENTRY IS MADE HERE FROM THE ERDR ROUTINE TO SET THE WLR INDICATOR WHEN THE WLR COUNTER HAS OVERFLOWED DURING A READ OR SENSE COMMAND WITH THE SKIP FLAG ON.		

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ERCX 001	T			XFER FROM COLUMN FORM BUFFER(RO) TO CCW ADDR	
1B94	5E88	ERCX 002	READ			RDH H AS, I+2	RD 2 BYTES
1B96	6FF3	ERCX 003				H1=H1+H1	
1B98	6FF3	ERCX 004				H1=H1+H1	SHIFT 8,9 TWICE
1B9A	5FB3	ERCX 005				T1=H1XH	T1=8,9
1B9C	6BE5	ERCX 006				T1=T1\$H0	T1=8,9,12,11,0,1,2,3
1B9E	1B73	ERCX 007				T1=T1*-K07	T1=8,9,12,11,0
1BA0	5FF9	ERCX 008				H1=H1	H1=4,5,6,7,8,9
1BA2	E0A6	ERCX 009		011	NOHI	BR IF HZNZ	BR IF 4,5,6,7 NO BIT
1BA4	9A73	ERCX 010		057	CHECK N	N=H0 BITS567	MASK ON BITS 1,2,3
1BA6	CE83	ERCX 011	NOHI			Z=H0*-K08	BITS 1,2,3
1BA8	F0B4	ERCX 012		018	GOVDTY	BR IF LZNZ	BR IF ANY IN 1,2,3
1BAA	3B43	ERCX 013				T1=T1\$K04	AT LEASE 4
1BAC	CF31	ERCX 014		016	BIT4ON	BR IF H1 BIT0=1	BR IF BIT 4
1BAE	5F73	ERCX 015		057	CHECK N	N=H1 BITS123	MASK ON BITS 5,6,7
1BB0	0F85	ERCX 016	BIT4ON			Z=H1*-K80	4 BIT ON
1BB2	E0DF	ERCX 017		021	XLATE	BR IF HZ=0	BR IF NO OTHER BIT
1BB4	59EE	ERCX 018	GOVDTY	067	CHECK 7	BR	
1BDA	3B23	ERCX 019	ORIN2			T1=T1\$K02	OR IN 2 BIT
1BDC	3B13	ERCX 020	ORIN1			T1=T1\$K01	OR IN 1 BIT
1BDE	5FA0	ERCX 021	XLATE			RDB H1 AS, T+0	XLATE TABLE LOOK UP
1BE0	7F38	ERCX 022				STB H1 V+1	SEND TO MAIN STORE
1BE2	21FF	ERCX 023	DCRCT			U1=U1+KFF	DECREMENT CCW CT
1BE4	C4B9	ERCX 024		032	CDFLAG	BR IF Z=0	
1BE6	261B	ERCX 025	START			DO=DO+K01	WLR CTR + 1
1BE8	F4B7	ERCX 026		031	WLRSET	BR IF AC=1	BR IF WLR
1BEA	E414	ERCX 027		002	READ	BR IF GO BIT2=0	BR IF NOT COL BIN
1BEC	5F88	ERCX 028				RDB H1 AS, I+1	COL BIN
1BEE	7F38	ERCX 029				STB H1 V+1	MOVE TO MAIN STORE
1BF0	9BE2	ERCX 030		023	DCRCT	BR	
1BB6	3C13	ERCX 031	WLRSET			PO=PO\$K01	WLR INDICATOR
1BB8	5629	ERCX 032	CDFLAG			VO=DO	MOVE WLR CTR
1BBA	2607	ERCX 033				DO=0	RST DO

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1BBC	54E9	ERCX 034				HO=GO	MOVE CMD BACK TO GO
1BBE	5460	ERCX 035				RDH G AS,D+0	RESTORE CHNL STATUS
1BC0	C548	ERCX 036		040	CDANOT	BR IF G1 BIT0=0	BR IF NOT CDA
1BC2	1525	ERCX 037				G1=G1*-K20	OVER COUNT, SUPPRESS SLI
1BC4	F84F	ERCX 038		043	PFRBRK	BR IF P0 BIT7=1	BR IF WLR
1BC6	9990	ERCX 039		ECOL 069	GOCHAN	BR	CDA
1BC8	2218	ERCX 040	CDANOT			VO=VO*K01	INCREMENT WLR CTR
1BCA	F4CF	ERCX 041		043	PFRBRK	BR IF AC=1	
1BCC	3C13	ERCX 042				PO=PO\$K01	WLR
1BCE	71A2	ERCX 043	PFRBRK			STB U1 DA,AC	STORE RESIDUE CT
1BD0	F70F	ERCX 045		050	ENPFR	BR IF D1 BIT3=1	BR IF NOT PFR
1BD2	CB06	ERCX 046		048	NOVAL	BR IF PS BIT4=0	PFR READ,BR IF NO VAL CHK(RU031)
1BD4	3D83	ERCX 047				P1=P1\$K08	VAL CHK,OR BIT IN SENSE
1BD6	CD80	ERCX 048	NOVAL			RST RP K=08	RST PCH CMD INTERLOCK
1BD8	9112	ERCX 049		EXFR 118	COMBIN	BR	
1B8E	8964	ERCX 050	ENPFR	ERRQ 102	STIND	BAL	STORE INDICATOR AND SENSE
1B90	7460	ERCX 054				STH G AS,D+0	STORE POSSIBLY MODIFIED SLI FLAG
1B92	988C	ERCX 055		ERDR 114	STACKR	BR	
19DE	3B23	ERCX 056	ORBIT2			T1=T1\$K02	OR IN 2 BIT
19E0	9BDE	ERCX 057	CHECK 0	021	XLATE	BR	
19E2	9BDA	ERCX 058	CHECK 1	019	ORIN2	BR	3 OR 7 BIT
19E4	99DE	ERCX 059	CHECK 2	056	ORBIT2	BR	2 OR 6 BIT
19E6	99EE	ERCX 060	CHECK 3	067	CHECK 7	BR	
19E8	9BDC	ERCX 061	CHECK 4	020	ORIN1	BR	1 OR 5 BIT
19EA	99EE	ERCX 062	CHECK 5	067	CHECK 7	BR	
19EC	99EE	ERCX 063	CHECK 6	067	CHECK 7	BR	
19EE	F775	ERCX 067	CHECK 7	070	RDVDCK	BR IF D1 BIT3=1	VALIDITY CHECK
19F0	2F80	ERCX 068				SET P K=08	SET PFR VAL CHK(RU031)
19F2	9BDE	ERCX 069		021	XLATE	BR	
19F4	2B80	ERCX 070	RDVDCK			SET R K=08	SET RDR VAL CHK(RU023)
19F6	9BDE	ERCX 072		021	XLATE	BR	
		ERCX 073	*			NATIVE 2540 READER TRANSLATE TABLE	
		ERCX 074	ATABLE	ADDR=3000			
3000	40F1	ERCX 075	C			X AUX' 40F1F2F3F4F5F6F7F061E2E3E4E5E6E760D1D2D3D4D5'	
3016	D6D7	ERCX 076	C			X AUX' D6D7D0A1A2A3A4A5A6A750C1C2C3C4C5C6C7C0818283'	
302C	8485	ERCX 077	C			X AUX' 848586876A9192939495969770B1B2B3B4B5B6B7F931'	
3042	3233	ERCX 078	C			X AUX' 323334353637E921222324252627D911121314151617'	
3058	A9E1	ERCX 079	C			X AUX' A9E1626364656667C901020304050607894142434445'	
306E	4647	ERCX 080	C			X AUX' 46479951525354555657B971727374757677F8797A7B'	
3084	7C7D	ERCX 081	C			X AUX' 7C7D7E7FE869E06B6C6D6E6FD8595A5B5C5D5E5FA8A0'	
309A	AAAB	ERCX 082	C			X AUX' AAABACADAEAF8494A4B4C4D4E4F88808A8B8C8D8E8F'	
30B0	9890	ERCX 083	C			X AUX' 98909A9B9C9D9E9F88B0BABBBCBDBEBF38393A3B3C3D'	
30C6	3E3F	ERCX 084	C			X AUX' 3E3F28292A2B2C2D2E2F18191A1B1C1D1E1F6820EAEB'	
30DC	ECED	ERCX 085	C			X AUX' ECEDEEEFF08090A030C0D0E0F4800CACBCCDCCECF5810'	
30F2	DADB	ERCX 086	C			X AUX' DADBDCDDDDDF7830FAFBFCFDFEFF'	
		ERCX C87	AEND				

 * CROSS REFERENCE FOR CSECT ERCX *

ERCX 002 ERCX 027
 ERCX 011 ERCX 009
 ERCX 016 ERCX 014
 ERCX 018 ERCX 012

* CROSS REFERENCE FOR CSECT ERCX *

ERCX 019	ERCX 058			
ERCX 020	ERCX 061			
ERCX 021	ERCX 017	ERCX 057	ERCX 069	ERCX 072
ERCX 023	ERCX 030			
ERCX 025	ERDR 169			
ERCX 031	ERCX 026	ERDR 172		
ERCX 032	ERCX 024	ERDR 178		
ERCX 040	ERCX 036			
ERCX 043	ERCX 038	ERCX 041		
ERCX 048	ERCX 046			
ERCX 050	ERCX 045			
ERCX 056	ERCX 059			
ERCX 057	ERCX 010	ERCX 015		
ERCX 067	ERCX 018	ERCX 060	ERCX 062	ERCX 063
ERCX 070	ERCX 067			

ERDR DESCRIPTIVE TEXT

ENTRY POINTS

<p>NRDR ENTRY TO THIS LABEL IS FROM THE DCLB ROUTINE FOR A READER OR PUNCH START I/O, OR FOR A CHAINING OPERATION. A CHECK IS MADE FOR IPL, AND THE AVAILABILITY OF THE READER OR THE PUNCH.</p> <p>AVANOT ENTRY IS MADE HERE FROM THE ERRQ ROUTINE WHEN THE 2540 IS FOUND NOT AVAILABLE. CONDITION CODE 3 IS SET, THE ACTIVE AND SECONDARY BITS ARE RESET AND THE DPTQ ROUTINE IS BRANCHED TO FOR A UCW STORE.</p> <p>STACKR ENTRY IS MADE HERE FROM THE ERCX ROUTINE TO SET STACKER SELECT BITS, SET NOT-READ, AND STORE INDICATOR AND SENSE BYTES.</p> <p>SSDONE ENTRY IS MADE TO THIS LABEL FROM THE ERRQ ROUTINE TO SET THE FEED INDICATOR, AND SET UP AND ISSUE A FEED COMMAND WITH STACKER SELECT.</p>	<p>UEXPFR ENTRY IS FROM THE EPCH ROUTINE WHEN A PFR UNIT EXCEPTION IS DETECTED. SET UNIT EXCEPTION IN STATUS REGISTER AND RESET COMMAND INTERLOCK. IF IPL, GO TO THE ERRQ ROUTINE AND STOP. IF NOT IPL, GO TO THE DPTS ROUTINE TO STORE STATUS AND FLAGS IN THE UCW.</p> <p>CMDREJ ENTRY IS MADE HERE FROM THE EPCH ROUTINE WHEN AN INVALID COMMAND DECODE, OR AN INVALID COMMAND SEQUENCE IS DETECTED.</p> <p>BACK THE EPCH ROUTINE BRANCHES HERE TO SETUP THE BEGINNING ADDRESSES OF THE COLUMN IMAGE BUFFER AND TRANSLATE TABLE MODULE.</p> <p>BRSKIP ENTRY HERE IS FROM THE EXFR ROUTINE AFTER THE PFR COLUMN BUFFER ADDRESS IS SETUP. IF THE SKIP FLAG IS ON, DO SKIP INDICATED. IF SKIP FLAG OFF, GO TO THE ERCX ROUTINE TO START TRANSLATE OF COLUMN IMAGE TO BYTE CHARACTERS FOR PROGRAM STORAGE.</p>
---	---

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ERDR 001	T			NATIVE 2540 RDR OR PCH SID OR CHAIN DECODE ROUTINE	
		ERDR 002	*				
		ERDR 003	*				
		ERDR 004	*			ENTRY TO THIS ROUTINE IS MADE FROM DCLB TO LABEL NRDR OF THIS ROUTINE.	
		ERDR 005	*			THE REGISTERS CONTAIN THE FOLLOWING INFO	
		ERDR 006	*				
		ERDR 007	*				
		ERDR 008	*			COUNT	U
		ERDR 009	*			DATA ADDRESS	V
		ERDR 010	*			COMMAND BYTE	G0
		ERDR 011	*			FLAG	G1
		ERDR 012	*				
		ERDR 013	*				
		ERDR 014	*			NATIVE 2540 READER UCW FORMAT(AUX STORAGE)	
		ERDR 015	*				
		ERDR 016	*			00DB	00DA
		ERDR 017	*			00DC	00DE
		ERDR 018	*			*-----*	
		ERDR 019	*			* CANNEL *CHANNEL *UNIT *NOT * RESIDUAL * NEXT CCW *	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ERDR 020	*			* STATUS* FLAGS	* STATUS * USED * COUNT * ADDRESS *
		ERDR 021	*			* * * * *	* * * * *
		ERDR 022	*			-----	-----
		ERDR 023	*				
		ERDR 024	*			CH STAT-D8	* CH FLAGS-D9 * UNIT STAT-DA
		ERDR 025	*			0-SECONDARY	* 0-CHAIN DATA * 0-NOT USED
		ERDR 026	*			1-WLR	* 1-CHAIN CMD * 1-NOT USED
		ERDR 027	*			2-PROG CHK	* 2-SLI * 2-NOT USED
		ERDR 028	*			3-PROT CHK	* 3-SKIP * 3-BUSY
		ERDR 029	*			4-NOT USED	* 4-PCI * 4-CH END
		ERDR 030	*			5-NOT USED	* 5-ACTIVE * 5-DEV END
		ERDR 031	*			6-INTFCE CK	* 6-NOT USED * 6-UNIT CHK
		ERDR 032	*			7-INT IN BFR	* 7-NOT USED * 7-UNIT EXCEPT
		ERDR 033	*				
		ERDR 034	*				
		ERDR 035	*			*****	*****
		ERDR 036	*			* INDICATORS AND SENSE BYTE *	
		ERDR 037	*			*****	*****
		ERDR 038	*			-----	-----
		ERDR 039	*			* INDICATORS	* SENSE *
		ERDR 040	*			* AUX STORAGE	* AUX STORAGE *
		ERDR 041	*			* ADDRESS	* ADDRESS *
		ERDR 042	*			* 10F0	* 10F1 *
		ERDR 043	*			-----	-----
		ERDR 044	*			* 0-CHAIN DATA	* 0- CMD REJECT *
		ERDR 045	*			* 1-STACKER SEL	* 1- NOT READY *
		ERDR 046	*			* 2-STACKER SEL	* 2- NOT USED *
		ERDR 047	*			* 3-0	* 3- EQUIP CHK *
		ERDR 048	*			* 4-READ	* 4-VALIDITY CHK *
		ERDR 049	*			* 5-COUNT STORED	* 5- NOT USED *
		ERDR 050	*			* 6-FEED	* 6-UNUSUAL CMD *
		ERDR 051	*			* 7-WLR	* 7- NOT USED *
		ERDR 052	*			-----	-----
		ERDR 053	ASEQ	AL07=12		HO=GO	MOVE CMD INTO HO
1812	54E9	ERDR 054	NRDR		060	NOTCDA	BR IF ZNZ
1814	C4A0	ERDR 055					BR IF NOT DATA CHAIN
1816	5EB2	ERDR 056					CDA, RESTORE OLD CMD INTO HO
1818	5D69	ERDR 057					RESTORE WLR CTR
181A	A044	ERDR 058			ERRQ 092	RDIND	BAL
181C	3C85	ERDR 059					READ INDICATORS & SENSE INTO P
181E	930E	ERDR 060					SET CDA INDICATOR
1820	C3AE	ERDR 061	NOTCDA	EPCH 057	067	HIGHCT	GO TO ADJUST HIGH COUNT
1822	F72C	ERDR 062			066	NOTIPL	BR IF BA BIT4=0
1824	24C6	ERDR 063					BR IF NOT IPL
1826	CEAC	ERDR 064					BR IF IPL FROM PUNCH
1828	3D23	ERDR 065			066	GOSTOP	SET MODE K=30
182A	DDAF	ERDR 066					SET READER MODE
182C	5EE0	ERDR 067					BR IF RPS BIT0=0
182E	8DC0	ERDR 068					BR IF OFF LINE(SX041)
1830	A044	ERDR 069					SET WAIT BIT FOR IPL CLUTCHING
1832	CEBD	ERDR 070			067	NOTIPL	BR IF RDR READY(SX041)
1834	9258	ERDR 071	GCSTOP	ERRQ 048		STOP	BR **IPL NOT AVAILABLE OR NOT READY,OR IPL FROM PCH
1836	2C75	ERDR 072	NOTIPL	CCOM 056		LSAVE	SIO OR CC, SAVE U, G, I & P
1838	4452	ERDR 073	AVANOT	ERRQ 092		RDIND	PO=INDICATORS, P1=SENSE
					074	AVABLE	BR IF ON LINE(SX041)
					CCOM 066	LRSTRB	RESTORE P AND I
							NOT AVAILABLE, SET COND. CODE 3
							G=0000, RST ACTIVE, SECONDARY

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
183A	S054	ERDR 073		DPTQ 020	STINST	BR	STORE UCW & GO TO BSWI
183C	5460	ERDR 074	AVABLE			RDH G AS, D+0	RD CH.ST. SO LATER CAN STORE UCW
183E	F747	ERDR 075		079	RDR1	BR IF D1 BIT3=1	BR IF RDR
1840	2D80	ERDR 076				SET RP K=08	SET PCH CMD INLK(RU031)
1842	0F30	ERDR 077				RST P K=03	RST PCH REQUEST IF HWDE(RU015)
1844	984A	ERDR 078		081	H10A	BR	
1846	2D04	ERDR 079	RDR1			SET RP K=20	SET RDR CMD INLK(RU021)
1848	0B30	ERDR 080				RST R K=03	RST RDR REQUEST IF HWDE(RU015)
184A	2FA3	ERDR 081	H10A			H1=0\$K0A	SET H1 TO LATER RST INDICATORS
184C	0E3B	ERDR 082				Z=H0\$K03	MASK FOR NO OP
184E	C4DC	ERDR 083		090	ENNOOP	BR IF ZNZ	BR IF NOT NO OP
1850	F758	ERDR 084		088	PCH1	BR IF D1 BIT3=0	NO OP, BR IF PCH
1852	DDDA	ERDR 085		089	NOTRDY	BR IF RS BIT1=0	BR IF RDR NOT READY(SX041)
1854	B16A	ERDR 086	RDYNOP	ERRQ 110	GUESS	BAL	RST CMD INTERLOCK
1856	861C	ERDR 087		DPTC 021	NOOP	BR	SHARE 1403 NO OP ROUTINE
1858	CFD5	ERDR 088	PCH1	086	RDYNOP	BR IF PS BIT0=1	BR IF PCH READY(SX041)
185A	98B6	ERDR 089	NOTRDY	139	UNITCK	BR	NOT READY
185C	0E4B	ERDR 090	ENNOOP			Z=H0\$K04	MASK FOR SENSE CMD
185E	F76F	ERDR 091		099	RDR2	BR IF D1 BIT3=1	BR IF RDR
1860	F0E9	ERDR 092		096	PCHSNS	BR IF LZ=0	BR IF PCH SENSE CMD
1862	2DC7	ERDR 093				P1=0	NOT SENSE CMD, ZERO OUT SENSE
1864	CFDA	ERDR 094		089	NOTRDY	BR IF PS BIT0=0	BR IF PCH NOT READY(SX041)
1866	92D4	ERDR 095		EPCH 010	NPCH	BR	GO TO PCH CMD DECODE
1868	CFF3	ERDR 096	PCHSNS	101	SNSRDY	BR IF PS BIT0=1	BR IF PCH READY(SX041)
186A	3D45	ERDR 097	SNSNRY			P1=P1\$K40	SENSE CMD, NOT READY
186C	9874	ERDR 098		102	JOIN	BR	
186E	F0FA	ERDR 099	RDR2	105	NOTSNS	BR IF LZNZ	BR IF NOT SENSE CMD
1870	DDEA	ERDR 100		097	SNSNRY	BR IF RS BIT1=0	RDR SNS CMD, BR IF NOT RDY(SX041)
1872	1D45	ERDR 101	SNSRDY			P1=P1*-K40	SENSE CMD, READY
1874	B16A	ERDR 102	JOIN	ERRQ 110	GUESS	BAL	RST CMD INTERLOCK
1876	3623	ERDR 103				D0=D0\$K02	SET 6 BIT FOR 1403 TO BR ON
1878	A4E8	ERDR 104		DYPE 165	GOCAL	BR	GO TO PICK UP H1=P1 WORD
187A	2DC7	ERDR 105	NOTSNS			P1=0	RDR. NOT SNS CMD, ZERO OUT SNS
187C	DDDA	ERDR 106		089	NOTRDY	BR IF RS BIT1=0	BR IF NOT RDY(SX041)
187E	1B00	ERDR 107				RST R K=80	RST RD & VAL CHK(RU023)
1880	6CF7	ERDR 108				PO=PO*H1	RST ALL IND. EXCEPT FD IND.
1882	EA34	ERDR 109		138	CMDREJ	BR IF H0 BIT6=0	BR IF NO 6 BIT - CMD REJECT
1884	FA26	ERDR 110		127	RDORRF	BR IF H0 BIT7=0	BR IF RD OR RD-FD
1886	EE34	ERDR 111		138	CMDREJ	BR IF H0 BIT2=0	CMC REJ IF NO 2 BIT
1888	E835	ERDR 112		138	CMDREJ	BR IF P0 BIT6=1	FEED CMD, BR IF DOUBLE FD
188A	76F2	ERDR 113				STH D DA, BE	FD ONLY, IMMEDIATE CMD, SAVE D07
188C	CE1A	ERDR 114	STACKR	121	SS1OR2	BR IF H0 BIT0=0	BR IF STKR SEL 1 OR 2
188E	DE18	ERDR 115		120	RP3	BR IF H0 BIT1=0	BR IF RP3
1890	FA35	ERDR 116		138	CMDREJ	BR IF H0 BIT7=1	BR IF NOT RD, CMD REJECT
1892	0D04	ERDR 117				RST RP K=20	RST RDR CMD INLK(RU021)
1894	A078	ERDR 118		ERRQ 083	RDCHST	BAL	RESTORE CH. STATUS AND FLAGS
1896	910A	ERDR 119		EXFR 114	CHKCHK	BR	RD NO FD, FROM ERCX, NOT PFR
1898	3C45	ERDR 120	RP3			PO=PO\$K40	STKR SELECT RP3
189A	DE1E	ERDR 121	SS1OR2	123	SSDCNE	BR IF H0 BIT1=0	BR IF R1 OR RP3
189C	3C25	ERDR 122				PO=PO\$K20	STKR SELECT R2
189E	1C83	ERDR 123	SSDONE			PO=PO*-K08	RST READ INDICATOR
18A0	3C23	ERDR 124				PO=PO\$K02	SET FEED INDICATOR
18A2	8964	ERDR 125		ERRQ 102	STIND	BAL	STORE INDICATORS & SENSE

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
18A4	9C03	ERDR 126		180	SETSKR N	N=PO BITS123	ISSUE FD CMD & STKR SELECT
18A6	DA35	ERDR 127	RDORRF	138	CMDREJ	BR IF HO BIT5=1	CMD REJECT IF 5-6 COMBINATION
18A8	EDAF	ERDR 128		131	UNITEX	BR IF RS BIT2=1	BR IF UNIT EXCEPTION(SX042)
18AA	2B40	ERDR 129				SET R K=04	RD OR RD-FD CMD,SENSE SET(RU019)
18AC	9304	ERDR 130		EPCH 052	PICKUP	BR	**SHARE UNUSUAL CMD SEQ BR & COUNT ADJUST
18AE	3D00	ERDR 131	UNITEX			SET RP K=80	RST EOF LITE(RU023)
18B0	2E13	ERDR 133	UEXPFR			HO=0\$K01	UNIT EXCEPTION IN UNIT STAT
18B2	98B8	ERDR 137		140	GOSTIN	BR	
18B4	2D85	ERDR 138	CMDREJ			P1=0\$K80	CMD REJ IN SENSE
18B6	2E23	ERDR 139	UNITCK			HO=0\$K02	UNIT CK IN UNIT STAT OF CSW
18B8	8964	ERDR 140	GOSTIN	ERRQ 102	STIND	BAL	STORE P - IND, & SENSE
18BA	B16A	ERDR 141		ERRQ 110	GUESS	BAL	RST CMD INTERLOCK
18BC	2A07	ERDR 142				TO=0	RST TO IN CASE OF CDA FLAG ON
18BE	C3C2	ERDR 143		145	IPLNOT	BR IF BA BIT4=0	BR IF NOT IPL
18C0	9EE0	ERDR 144		ERRQ 048	STOP	BR	IPL CMD REJ OR UNIT EXCEPTION
18C2	9D3C	ERDR 145	IPLNOT	DPTS 066	PCENTR	BR	
18C4	CC5D	ERDR 146	BACK	162	MOVEOP	BR IF PO BIT0=1	PICKED UP CDA & PFR RD, BR IF CDA
18C6	F753	ERDR 148		154	ENPFR	BR IF D1 BIT3=1	SIO OR CC, BR IF NOT PFR-RD
18C8	2445	ERDR 149				GO=0\$K40	PFR-RD SIO OR CC
18CA	2513	ERDR 150				G1=0\$K01	G=4001 TO PFR COL BUFR
18CC	40A6	ERDR 151				U=T	U=10F2
18CE	31BB	ERDR 152				U1=U1-K0B	**U=10E6 FOR LAST 10 BYTES IN IMAGE BUFR
18D0	905A	ERDR 153		EXFR 012	BYTECT	BR	PFR-RD. XFER FIRST & THEN RD
18D2	EE56	ERDR 154	ENPFR	159	NOTCOL	BR IF HO BIT2=0	BR IF NOT COL BIN
18D4	36A9	ERDR 158				DO=0-KA0	COL BIN. WLR CTR=5F
18D6	2855	ERDR 159	NOTCOL			IO=0\$K50	
18D8	2915	ERDR 160				I1=0\$K10	
18DA	3983	ERDR 161				I1=I1\$K08	I=5018, COL FORM BUFR ADDR
18DC	5E49	ERDR 162	MOVEOP			GO=HO	MOVE CMD TO GO FOR ERCX ROUTINE
18DE	F565	ERDR 164	BRSKIP	170	SKIP	BR IF G1 BIT3=1	BR IF SKIP FLAG ON
18E0	2A35	ERDR 168				TO=0\$K30	XLATE TABLE ADDR
18E2	9BE6	ERDR 169		ERCX 025	START	BR	**GO TO RD, COL OR NORMAL, OR PFR.SIO,CC,OR CDA
18E4	261B	ERDR 170	SKIP			DO=DO+K01	WLR CTR + 1
18E6	F4EA	ERDR 171		173	IADD1	BR IF AC=0	BR IF NO QVER CT
18E8	9BB6	ERDR 172		ERCX 031	WLRSET	BR	SKIPPED TOO MANY, WLR
18EA	5884	ERDR 173	IADD1			I=I+1	ADD 1 TO COL BUFR ADDR
18EC	E471	ERDR 174		176	CTADD1	BR IF GO BIT2=1	BR IF COL BIN
18EE	5884	ERDR 175				I=I+1	TWO BYTES OF COL FORM PER BYTE
18F0	21FF	ERDR 176	CTADD1			U1=U1+KFF	ADJUSTED CCW CT LESS 1
18F2	C4E4	ERDR 177		170	SKIP	BR IF ZNZ	BR IF NOT DONE SKIPPING
18F4	9BB8	ERDR 178		ERCX 032	CDFLG	BR	FINISH SKIPPING
		ERDR 179	AEND				
1900	2BC8	ERDR 180	SETSKR 0			SET R K=40	PKT R1(RU021)
1902	9050	ERDR 181		EXFR 007	XFER	BR	GO TO XFER ROW FORM TO COL BUFR
1904	2BCC	ERDR 182	SETSKR 2			SET R K=60	PKT R2(RU021 & 023)
1906	9050	ERDR 183		EXFR 007	XFER	BR	
1908	2B0A	ERDR 184	SETSKR 4			SET R K=50	PKT RP3(RU021 & 023)
190A	9050	ERDR 185		EXFR 007	XFER	BR	

 * CROSS REFERENCE FOR CSECT ERDR *

ERDR 053	DCLB 211	DCLB 212							
ERDR 060	ERDR 054								
ERDR 066	ERDR 061	ERDR 063							
ERDR 067	ERDR 060	ERDR 065							
ERDR 071	ERRQ 007								
EKDR 074	ERDR 069								
ERDR 079	ERDR 075								
ERDR 081	ERDR 078								
EKDR 086	ERDR 088								
ERDR 088	EKDR 084								
ERDR 089	ERDR 085	ERDR 094	ERDR 106						
EKDR 090	ERDR 083								
ERDR 096	ERDR 092								
EKDR 097	ERDR 100								
ERDR 099	ERDR 091								
ERDR 101	ERDR 096								
ERDR 102	ERDR 098								
ERDR 105	ERDR 099								
ERDR 114	ERCX 055								
ERDR 120	ERDR 115								
ERDR 121	ERDR 114								
ERDR 123	ERDR 121	ERRQ 069							
ERDR 127	ERDR 110								
EKDR 131	ERDR 128								
ERDR 133	EPCH 050								
EKDR 138	EPCH 032	ERDR 109	ERDR 111	ERDR 112	ERDR 116	ERDR 127			
ERDR 139	ERDR 089								
ERDR 140	ERDR 137								
ERDR 145	ERDR 143								
ERDR 146	EPCH 071								
ERDR 154	ERDR 148								
EKDR 159	ERDR 154								
EKDR 162	ERDR 146								
ERDR 164	EXFR 099								
ERDR 170	ERDR 164	ERDR 177							
EKDR 173	ERDR 171								
ERDR 176	ERDR 174								
ERDR 180	ERDR 126								

ERRQ DESCRIPTIVE TEXT

ENTRY POINTS

REDUC	ENTRY IS FROM THE DCLA ROUTINE WHEN A START I/O OR TEST I/O IS DECODED (FOR READER OR PUNCH), AND DEVICE END HAS BEEN CLEARED.	STOP	ENTRY IS FROM THE DCLR ROUTINE WHEN A CHANNEL ERROR IS DETECTED DURING AN IPL. ENTRY IS FROM THE ERDR ROUTINE WHEN THE 2540 IS NOT AVAILABLE FOR IPL, NOT READY, IPL FROM PUNCH ATTEMPTED, IPL COMMAND REJECT, OR A UNIT EXCEPTION DETECTED.
PCHST	ENTRY IS FROM THE BSWI ROUTINE WHEN A PUNCH STATUS REQUEST HAS BEEN DETECTED. READOUT THE UCW BYTES INTO LOCAL STORAGE, RESET READER STATUS AND DEVICE END.	RDCONT RDCHST RPHUCW RDIND STIND GUESS	THESE ARE COMMON ENTRY POINTS BRANCHED TO BY BRANCH AND LINK WORDS TO PERFORM SOME OF THE BASIC FUNCTIONS FOR THE 2540.
RDST	ENTRY IS FROM THE BSWI ROUTINE WHEN A READER STATUS REQUEST HAS BEEN DETECTED. READOUT THE UCW BYTES INTO LOCAL STORAGE, RESET READER STATUS AND DEVICE END.		

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ERRQ 001	T			2540 REQUEST & TEST I/O ROUTINE	
		ERRQ 002	*				
1EAA	AC6E	ERRQ 004	REDUC	077	RDUUCW	BAL	READ CH.STAT, FLAGS, & UNIT STAT
1EAC	3623	ERRQ 005				DO=DO\$K02	SET DO 6 BIT TO BE BR ON LATER
1EAE	CEB3	ERRQ 006		008	AVABLE	BR IF RPS BIT0=1	BR IF AVABLE(SX041)
1FB0	9836	ERRQ 007		ERDR 071	AVANOT	BR	NOT AVAILABLE
1EB2	F712	ERRQ 008	AVABLE	012	PCH1	BR IF D1 BIT3=0	BR IF PCH
1EB4	C43D	ERRQ 009		030	PCKUPD	BR IF GO BIT0=1	BR IF SECONDARY ON, RDR
1EB6	DD96	ERRQ 010		014	NOTRDY	BR IF RS BIT1=0	BR IF RDR NOT RDY(SX041)
1EB8	B362	ERRQ 011	RDY	CCOM 154	CCOB	BR	BR TO SET COND. CODE 0
1E92	C46D	ERRQ 012	PCH1	017	PICKUP	BR IF GO BIT0=1	BR IF SECONDARY ON, PCH
1E94	CFB9	ERRQ 013		011	RDY	BR IF PS BIT0=1	BR IF PCH RDY(SX041)
1E96	E954	ERRQ 014	NOTRDY	DPTT 013	UCENT	BR	NOT READY
		ERRQ 015	*			2540 REQUEST	
1EEA	A062	ERRQ 016	PCHST	087	RPHUCW	BAL	**PCH REQ. RD CH.ST, FLAG, UNIT STAT, CT
1EEC	DB98	ERRQ 017	PICKUP	054	NODE	BR IF PS BIT5=0	BR IF NO PCH HWDE(SX043)
1EEE	0F10	ERRQ 018				RST P K=01	RST PCH HWDE(RU015)
1EF0	C446	ERRQ 019		035	SETSEC	BR IF GO BIT0=0	BR IF NOT-READY TO READY
1EF2	FFC8	ERRQ 020		036	CCBR	BR IF PS BIT3=0	BR IF NO PCH CHK (SX042)
1EF4	8DC4	ERRQ 021		CCOM 058	LSAVEB	BAL	PCH CK, SAVE P, I
1EF6	2D15	ERRQ 022				P1=0\$K10	EQUIPMENT CK IN SENSE
1EF8	5CCD	ERRQ 023				PO=POL	RETAIN PFR RD & PFR WR INDICATOR
1EFA	8964	ERRQ 024		102	STIND	BAL	STORE INDICATORS & SENSE
1EFC	7CA0	ERRQ 025				STH P AS, T+0	STORE FORCED STKR SEL IN 10F2
1EFE	1545	ERRQ 026				G1=G1*-K40	BREAK CC
1F00	3E23	ERRQ 027				H0=H0\$K02	UNIT CK, DE IN UNIT STAT
1F02	9EC8	ERRQ 028		036	CCBR	BR	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1EBA	A06A	ERRQ 029	RDST	075	RRDU CW	BAL	**RDR REQ. RD CH. ST, FLAG, UNIT STAT, CT
1EBC	2466	ERRQ 030	PCKUPD			SET MODE K=36	RDR MODE, 2540 ZONE
1EBE	E607	ERRQ 031		064	CPUXFR	BR IF D0 BIT2=1	BR IF RUN-IN CPU XFER
1ECO	24C6	ERRQ 032				SET MODE K=30	SET BACK FIRST TO RDR MODE
1EC2	0998	ERRQ 033		054	NODE	BR IF RS BIT5=0	BR IF NO RDR HWDE(SX043)
1EC4	0B10	ERRQ 034				RST R K=01	RST RDR HWDE
1EC6	3485	ERRQ 035	SETSEC			GO=GO\$K80	**SET SECONDARY IN CASE OF QUE HWDE-READY
1EC8	811A	ERRQ 036	CCBR	DCOM 016	RSTREQ	BAL	RST RDR DR PCH REQUEST
1ECA	054E	ERRQ 037		039	NOCC	BR IF G1 BIT1=0	BR IF NO CMD CHAIN
1ECC	9D44	ERRQ 038		DPTQ 036	CHAIOK	BR	GO TO CHAIN
1ECE	3E43	ERRQ 039	NOCC			HO=HO\$K04	DE IN UNIT STAT
1ED0	E263	ERRQ 040		049	TIO	BR IF D0 BIT6=1	BR IF FROM TIO OR SIO
1ED2	F728	ERRQ 041		063	GOINT	BR IF D1 BIT3=0	BR IF PCH
1ED4	C3A8	ERRQ 042		063	GOINT	BR IF BA BIT4=0	BR IF NOT IPL
1ED6	D461	ERRQ 043		048	STOP	BR IF G0 BIT1=1	IPL, BR IF WLR IN CH. STAT
1ED8	EA61	ERRQ 044		048	STOP	BR IF H0 BIT6=1	BR IF UNIT CHECK
1EDA	4452	ERRQ 045				RDH G DC, 9A	G=0000, RST ACTIVE, SECONDARY
1EDC	A568	ERRQ 046		DCOM 049	REDSTR	BAL	STORE UCW
1EDE	8776	ERRQ 047		DCLL 036	PICKUP	BR	IPL DONE
1EE0	50C7	ERRQ 048	STOP			STOP	NATIVE 2540 IPL STOP
1EE2	2A07	ERRQ 049	TIO			TO=0	RST IO
1EE4	B11A	ERRQ 050		DCOM 016	RSTREQ	BAL	RST READER, PCH REQUEST
1EE6	D104	ERRQ 051		053	NOACT	BR IF G1 BIT5=0	BR IF NO ACTIVE BIT
1EE8	895E	ERRQ 052		DPTT 023	RESTES	BR	TIO OR SIO, ACT, SEC=11
1E84	8924	ERRQ 053	NOACT	DPTT 049	SHARE	BR	TIO, ACT, SEC=01
1E98	E263	ERRQ 054	NODE	049	TIO	BR IF D0 BIT6=1	BR IF TIO OR SIO
1E9A	C3C9	ERRQ 055		036	CCBR	BR IF BA BIT4=1	BR IF IPL
		ERRQ 056	*				INTERFACE CHECK OR UNQUEUED
1E9C	C424	ERRQ 057		061	INTFCK	BR IF G0 BIT0=0	BR IF NO SECONDARY
1E9E	D129	ERRQ 058		063	GOINT	BR IF G1 BIT5=1	BR IF ACTIVE
1EA0	CA25	ERRQ 059		061	INTFCK	BR IF H0 BIT4=1	NOT ACTIVE, BR IF NO CHE
1EA2	DA29	ERRQ 060		063	GOINT	BR IF H0 BIT5=1	BR IF DE
1EA4	3423	ERRQ 061	INTFCK			GO=GO\$K02	INTERFACE CK IN CHNL STAT
1EA6	15C5	ERRQ 062				G1=G1*-KCO	BREAK CHAIN
1EA8	9D52	ERRQ 063	GOINT	DPTQ 019	TAKEIN	BR	GO FOR IB, STORE UCW, GO BSWI
1E86	2406	ERRQ 064	CPUXFR			SET MODE K=30	RDR MODE, CPU ZONE
1E88	0B30	ERRQ 065				RST R K=03	RST RDR REQUEST & HWDE(RU015)
1E8A	8DC4	ERRQ 066		CCOM 058	LSAVEB	BAL	SAVE P, I
1E8C	A044	ERRQ 067		092	RDIND	BAL	READ INDICATORS & SENSE
1E8E	2D04	ERRQ 068				SET RP K=20	SET READ COMMAND INTLK
1E90	989E	ERRQ 069		ERDR 123	SSDONE	BR	GO TO XFER BUFR
		ERRQ 071	*				
		ERRQ 072	*				BRANCH AND LINK ROUTINES
		ERRQ 073	*				
206A	4642	ERRQ 075	RRDU CW			RDH D DC, 98	READ CONSTANT OF OOC8
206C	3715	ERRQ 076				D1=D1\$K10	D=0008
206E	F766	ERRQ 077	RDU CW	089	PHUCW	BR IF D13=0	BR IF PUNCH
2070	24C6	ERRQ 078				SET MODE K=30	SET READER MODE
2072	3743	ERRQ 080	RDCONT			D1=D1\$K04	D=CXXC
2074	5060	ERRQ 081				RDH U AS, D+0	READ COUNT INTO U
2076	1743	ERRQ 082				D1=D1*-K04	D=0XX8
2078	5468	ERRQ 083	RDCHST			RDH G AS, D+2	READ CHAN STAT FLAGS
207A	5E6A	ERRQ 084				RDH H AS, D-2	READ UNIT STAT

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
207C	128E	ERRQ 085				RTN	
2062	4642	ERRQ 087	RPHUCW			RDH D DC,98	READ CONSTANT OF 00C8
2064	3725	ERRQ 088				D1=D1\$K20	D=00E8
2066	2402	ERRQ 089	PHUCW			SET MODE K=10	SET PUNCH MODE
2068	C4F2	ERRQ 090		080	RDCONT	BR IF ZNZ	
		ERRQ 091					
2044	2A15	ERRQ 092	RDIND			T0=0\$K10	
2046	F75C	ERRQ 093		098	RPHIND	BR IF D1 BIT3=0	BR IF PCH
2048	2406	ERRQ 094				SET MODE K=30	RDR MODE
204A	2BF5	ERRQ 095				T1=0\$KFO	T=10F0
204C	5CAA	ERRQ 096	READ			RDH P AS,T-2	PO=INDICATORS, P1=SENSE
204E	128E	ERRQ 097				RTN	
2050	24C2	ERRQ 098	RPHIND			SET MODE K=10	PCH MODE
2052	2BB1	ERRQ 099				T1=0-K0B	T=10F4
2054	C4CC	ERRQ 100		096	READ	BR IF ZNZ	UNCONDITIONAL BR
		ERRQ 101					
0964	2A15	ERRQ 102	STIND			T0=0\$K10	
0966	F76E	ERRQ 103		107	SPHIND	BR IF D1 BIT3=0	BR IF PH
0968	2BF5	ERRQ 104				T1=0\$KFO	T=10F0
096A	7CAA	ERRQ 105	STORE			STH P AS,T-2	STORE INDICATORS & SENSE
096C	128E	ERRQ 106				RTN	
096E	2BB1	ERRQ 107	SPHIND			T1=0-K0B	T=10F4
0970	C4EA	ERRQ 108		105	STORE	BR IF ZNZ	UNCONDITIONAL BR
		ERRQ 109					
316A	F772	ERRQ 110	GUESS	114	PCH2	BR IF D1 BIT3=0	BR IF PCH
316C	D271	ERRQ 111		113	RETUN	BR IF D0 BIT5=1	BR IF FEED ONLY
316E	0D04	ERRQ 112				RST RP K=20	RST RDR CMD INLK(RU021)
3170	128E	ERRQ 113	RETUN			RTN	
3172	0D80	ERRQ 114	PCH2			RST RP K=08	RST PCH CMD INLK(RU031)
3174	128E	ERRQ 115				RTN	

 * CROSS REFERENCE FOR CSECT ERRQ *

ERRQ 004	DCLA 305	DCLA 306					
ERRQ 008	ERRQ 006						
ERRQ 011	ERRQ 013						
ERRQ 012	ERRQ 008						
ERRQ 014	ERRQ 010						
ERRQ 016	BSWI 059						
ERRQ 017	ERRQ 012						
ERRQ 029	BSWI 056						
ERRQ 030	ERRQ 009						
ERRQ 035	ERRQ 019						
ERRQ 036	ERRQ 020	ERRQ 028	ERRQ 055				
ERRQ 039	ERRQ 037						
ERRQ 048	DCLR 034	ERDR 066	ERDR 144	ERRQ 043	ERRQ 044		
ERRQ 049	ERRQ 040	ERRQ 054					
ERRQ 053	ERRQ 051						
ERRQ 054	ERRQ 017	ERRQ 033					
ERRQ 061	ERRQ 057	ERRQ 059					
ERRQ 063	ERRQ 041	ERRQ 042	ERRQ 058	ERRQ 060			
ERRQ 064	ERRQ 031						
ERRQ 075	ERRQ 029	EXFR 109					

ETRP DESCRIPTIVE TEXT

ENTRY POINTS

(0110) THIS IS THE ASSIGNED TRAP ADDRESS THAT IS FORCED INTO THE SAR WHEN A 2540 PUNCH TRAP IS ALLOWED. DIAGRAM 5-115 IN THE FEMDM SHOWS THE READER TRAP HANDLING.

INLK ENTRY IS MADE HERE FROM THE BMCK ROUTINE. A MACHINE CHECK HAS BEEN DETECTED DURING A PUNCH TRAP, THE BMCK ROUTINE HAD BEEN TRAPPED TO, THE MMSK AND BA EXTERNALS HAVE BEEN SAVED AND THE LOGOUT LATCH HAS BEEN SET. THE OBJECT OF THIS ENTRY IS TO RESET COMMAND INTERLOCK AND RETURN TO THE MICROROUTINE TRAPPED FROM.

```

*****
*
*           PROCEDURE FOR ANALYZING PUNCH CHECKS DUE TO HOLE COUNT
*
* -1- SET ADDRESS SWITCHES TO ADDRESS OF LABEL -PCHCHK-
*
* -2- SET MODE SWITCH TO SAR DELAYED STOP
*
* -3- WHEN THE ERROR OCCURS, THE PUNCH ATTENTION LITE WILL COME ON AND THE FOLLOWING AREAS MAY BE DISPLAYED -
*
* ---PUNCH CHECK LOGOUT AREA---
*
*   AUX      50F6      50F7      50F8      50F9      50FA      50FB      50FC      50FD      50FE      50FF
*   *-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
*   BITS * 01234567 * 01234567 * 01234567 * 01234567 * 01234567 * 01234567 * 01234567 * 01234567 * 01234567 *
*   *           *           *           *           *           *           *           *           *           *
*   COLS * 1 THRU 8 * 9 THRU 16 * 17 THRU 24 * 25 THRU 32 * 33 THRU 40 * 41 THRU 48 * 49 THRU 56 * 57 THRU 64 * 65 THRU 72 *
*   *-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
*
*   A BIT BEING ON IN THE LOGOUT AREA SIGNIFIES
*   AN ERROR FOR THAT COLUMN.
*   THIS AREA MUST BE LOADED WITH BLANKS AFTER
*   EVERY ERROR FOR SUBSEQUENT ERROR ANALYSIS.
*
* ---ROW FORM PUNCH IMAGE---
*
*   ROW      AUX ADDRESS      ROW      AUX ADDRESS
*
*   12      400C - 4015      11      4016 - 401F
*   0       4020 - 4029      1       402A - 4033
*   2       4034 - 403D      3       403E - 4047
*   4       4048 - 4051      5       4052 - 405B
*   6       405C - 4065      7       4066 - 406F
*   8       4070 - 4079      9       407A - 4083
*
*   A BIT BEING OFF IN THE ROW FORM PUNCH IMAGE
*   BUFFER SIGNIFIES A HOLE IN THE CARD. THE CARD
*   IS IN THE STACKER POKET.
*
*****

```

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ETRP 001	T			NATIVE PUNCH TRAP ROUTINE	
		ETRP 002	ATABLE	ADDR=0110			
		ETRP 009	*				
		ETRP 010	*				
		ETRP 011	*			THE REGISTERS ARE USED FOR THE FOLLOWING	
		ETRP 012	*				
		ETRP 013	*			GO-TRAP COUNT	
		ETRP 014	*			V -AUX STORAGE ADDRESS MODIFICATION	
		ETRP 015	*			DO5=1 NORMAL WRITE INDICATOR	
		ETRP 016	*			DO7=1 12 ROW PUNCH	
		ETRP 017	*				
		ETRP 018	*				
		ETRP 019	*				
0110	22C8	ETRP 020				LINK U MMSK4=1	
0112	8FA4	ETRP 021		023	START	BR	
		ETRP 022	AEND				
OFA4	FAF4	ETRP 023	START	067	SETUP	BR IF RPS BIT7=0	BR IF CLUTCH TRAP(SX044)
OFA6	5ACF	ETRP 024				PO=RP2	GET PUNCH CHECK BR INFO
OFA8	5BFF	ETRP 025				H1=RP1	GET PFR INFORMATION
OFAA	5F2C	ETRP 026				RDB PO AS,V+1	SEND PUNCH DATA
OFAC	F25F	ETRP 027		056	ROW12	BR IF DO BIT7=1	BR IF 12TH ROW
OFAE	D232	ETRP 029		031	PFR	BR IF DO BIT5=0	BR IF PFR OPERATION
OFB0	DFC5	ETRP 030		040	NOTPFR	BR IF PS BIT1=1	BR IF NOT NORMAL RUN IN(SX041)
OFB2	236D	ETRP 031	PFR			V1=V1+K60	POINT TO
OFB4	23CB	ETRP 032				V1=V1+K0C	PFR AREA
OFB6	5D20	ETRP 033				RDB P1 AS,V+0	READ PFR ROW AREA
OFB8	7F20	ETRP 034				STB H1 AS,V+0	STORE NEW PFR INFO
OFBA	337F	ETRP 035				V1=V1-K77	POINT TO PCH IMAGE
OFBC	5F20	ETRP 036				RDB H1 AS,V+0	
OFBE	6DF7	ETRP 037				P1=P1*H1	OR PUNCH AND PFR DATA
OFCC	7D20	ETRP 038				STB P1 AS,V+0	STORE RESULTS
OFCE	23CB	ETRP 039				V1=V1+K0C	
OFD4	223D	ETRP 040	NOTPFR			VO=VO+K30	POINT ADDRESS TO CHECK AREA
OFD6	5F20	ETRP 044				RDB H1 AS,V+0	READ CHECK IMAGE BYTE
OFD8	6CF1	ETRP 045				PO=PO+H1	COMPARE RESULTS
OFDA	C4DD	ETRP 046		055	NOEROR	BR IF Z=0	BR IF NO ERROR
OFDC	CADC	ETRP 047		055	NOEROR	BR IF RPS BIT4=0	BR IF NO PCH CHK CL (SX044)
OFDE	3F40	ETRP 048	PCHCHK			SET P K=84	ERR,SET PCH CHK \$ RESTART(RU031)
OFD0	2E55	ETRP 049				HO=0\$K50	SET ERROR LOG OUT
OFD2	54FD	ETRP 050				H1=GOL	ADDRESS TO
OFD4	3FF5	ETRP 051				H1=H1\$KF0	50F6 - 50FF
OFD6	5DE0	ETRP 052				RDB P1 AS,H+0	READ OLD ERROR BYTE
OFD8	6DC5	ETRP 053				P1=P1\$P0	OR NEW ERROR TO OLD BYTE
OFDA	7DE0	ETRP 054				STB P1 AS,H+0	STORE NEW RESULTS
OFDC	2215	ETRP 055	NOEROR			VO=0\$K10	POINT BACK TO PCH IMAGE AREA
OFDE	241B	ETRP 056	ROW12			GO=GO+K01	UP DATE COUNT
OFE0	FOEA	ETRP 057		062	RETURN	BR IF LZNZ	BR IF NOT THE LAST TRAP OF A ROW
OFE2	F266	ETRP 058		060	GOADD6	BR IF DO BIT7=0	BR IF NOT 12TH ROW
OFE4	1613	ETRP 059				DO=DO*-K01	RST 12TH ROW INDICATOR
OFE6	246B	ETRP 060	GOADD6			GO=GO+K06	CORRECT GO TO X6
OFE8	EOED	ETRP 061		063	DONE	BR IF HZ=0	BR IF COUNT FINISHED
OFEA	E0C8	ETRP 062	RETURN			RTN U MMSK4=0	
OFEC	DFE2	ETRP 063	DONE	066	INLK	BR IF PS BIT1=0	FINISHED,BR IF NORM RUNIN(SX041)

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0FEE	CFF2	ETRP 064		066	INLK	BR IF PS BIT0=0	BR IFNOT READY(SX041)
0FF0	2F20	ETRP 065				SET P K=02	SET PCH REQUEST(RU015)
0FF2	0D80	ETRP 066	INLK			RST RP K=08	RST PCH CMD INLK(RU031)
0FF4	2215	ETRP 067	SETUP			V0=0\$K10	
0FF6	2307	ETRP 068				V1=0	V=1000
0FF8	5F2C	ETRP 069				RDB PO AS,V+1	PRE SHIP PUNCH DATA
0FFA	3613	ETRP 070				DO=00\$K01	SET 12TH ROW INDICATOR
0FFC	2435	ETRP 071				GO=0\$K30	SET COUNT TO 36 FOR 130
0FFE	8FE6	ETRP 072		060	GOADD6	BR	TRAPS IN DECIMAL

 * CROSS REFERENCE FOR CSECT ETRP *

ETRP 023	ETRP 021		
ETRP 031	ETRP 029		
ETRP 040	ETRP 030		
ETRP 055	ETRP 046	ETRP 047	
ETRP 056	ETRP 027		
ETRP 060	ETRP 058	ETRP 072	
ETRP 062	ETRP 057		
ETRP 063	ETRP 061		
ETRP 066	BMCK 042	ETRP 063	ETRP 064
ETRP 067	ETRP 023		

ETTR DESCRIPTIVE TEXT

ENTRY POINTS

(Q1B0) THIS IS THE ASSIGNED TRAP ADDRESS THAT IS FORCED INTO THE SAR WHEN A 2540 READ TRAP IS ALLOWED. DIAGRAM 5-106 IN THE FEMDM SHOWS THE READER TRAP HANDLING.

RQUEST ENTRY IS MADE TO THIS LABEL FROM THE BMCK ROUTINE. A MACHINE CHECK HAS BEEN DETECTED DURING A READER TRAP, THE BMCK ROUTINE HAD BEEN TRAPPED TO, THE MMSK AND BA EXTERNALS HAVE BEEN SAVED AND THE LOGOUT LATCH HAS BEEN SET. THE OBJECT OF THIS ENTRY IS TO SET A READER STATUS REQ AND RETURN TO THE ROUTINE TRAPPED FROM.

* PROCEDURE FOR ANALYZING READER CHECKS DUE TO HOLE COUNT *

* -1- SET ADDRESS SWITCHES TO ADDRESS OF LABEL -RDRCHK- *

* -2- SET MODE SWITCH TO SAR DELAYED STOP *

* -3- WHEN THE ERROR OCCURS, THE RDR ATTENTION LITE WILL COME ON AND THE FOLLOWING AREAS MAY BE DISPLAYED - *

* --READER CHECK LOGOUT AREA-- *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

* * * * * *

A BIT BEING ON IN THE LOGOUT AREA SIGNIFIES AN ERROR FOR THAT COLUMN. THIS AREA MUST BE LOADED WITH BLANKS AFTER EVERY ERROR FOR SUBSEQUENT ERROR ANALYSIS.

---COLUMN FORM AREA--- AUX LOCATIONS 5018 - 50B7
A BIT BEING ON INDICATES A HOLE IN THE CARD THAT IS NOW IN THE PRE STACKER STATION. THIS IS A RESULT OF THE FIRST READ STATION TRAPS.

FORMAT	5018		5019		
	BIT	ROW	COL		
	0	X			FORMAT IS THE SAME FOR EACH CARD COLUMN. AUX 501A-501B FOR COL 2, AUX 501C-501D FOR COL 3, ETC.
	1	X			
	2	12			
	3	11	-1-		
	4	0			
	5	1			
	6	2			
	7	3			

---ROW FORM AREA--- AUX LOCATIONS 4FA0 - 5017
A BIT BEING A ZERO INDICATES A HOLE IN THE CARD THAT IS BETWEEN THE FIRST AND SECOND READ STATIONS.

ROW	AUX ADDRESS	ROW	AUX ADDRESS
9	4FA0 - 4FA9	3	4FDC - 4FE5
8	4FAA - 4FB3	2	4FE6 - 4FEF
7	4FB4 - 4FBD	1	4FF0 - 4FF9
6	4FBE - 4FC7	0	4FFA - 5003
5	4FC8 - 4FD1	11	5004 - 500
4	4FD2 - 4FD8	12	500E - 5017

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ETTR 001	T			NATIVE READER TRAP ROUTINE	
		ETTR 002	ATABLE	ADDR=01B0			
		ETTR 009	*				
		ETTR 010	*				
		ETTR 011	*				
		ETTR 012	*			THE REGISTERS IN THIS ROUTINE ARE USED AS FOLLOWS	
		ETTR 013	*				
		ETTR 014	*			DO BIT2=1 XFER ROW TO COL	
		ETTR 015	*			D1 REG-TRAP COUNT	
		ETTR 016	*			U REG-ROW ADDRESS	
		ETTR 017	*				
01B0	2206	ETTR 018				LINK U MMSK3=1	
01B2	5ACF	ETTR 019				P0=RP2	GET READ 2 INFO
01B4	8ECC	ETTR 020		022	START	BR	
		ETTR 021	AEND				
0ECC	EAFC	ETTR 022	START	048	SETUP	BR IF RPS BIT6=0	BR IF SET-UP(SX044)
0ECE	5BDF	ETTR 023				P1=RP1	GET READ 1 INFO(START SHIFT REG)
0ED0	5F00	ETTR 024				RDB H1 AS,U+0	READ OLD DATA
0ED2	7D08	ETTR 025				STB P1 AS,U+1	STORE NEW DATA
0ED4	6CF1	ETTR 026				P0=P0H1	CHECK OLD READ 1 WITH NEW READ 2
0ED6	C4E9	ETTR 027		036	NOEROR	BR IF Z=0	BR IF NO ERROR
0ED8	CDE9	ETTR 028		036	NOEROR	BR IF RS BIT0=1	BR IF NOT GATE RD CMLTE(SX041)
0EDA	3800	ETTR 029	RDRCHK			SET R K=80	SET ERROR LAT (RU021)
0EEC	2E15	ETTR 030				H0=0\$K10	SET H TO ADDRESS READ CHECK
0EDE	57FD	ETTR 031				H1=D1L	LOG OUT AREA IN AUX STORAGE
0EE0	2FFD	ETTR 032				H1=H1+KFO	POSITION 10F6 THRU 10FF
0EE2	5DE0	ETTR 033				RDB P1 AS,H+0	READ OLD ERROR BYTE
0EE4	6DC5	ETTR 034				P1=P1\$P0	OR NEW ERROR
0EE6	7DE0	ETTR 035				STB P1 AS,H+0	STORE ORED RESULTS
0EE8	271B	ETTR 036	NOEROR			D1=D1+K01	UPDATE TRAP COUNT
0EEA	F0F0	ETTR 037		040	RETURN	BR IF LZNZ	
0EEC	2768	ETTR 038	D1ADD6			D1=D1+K06	CORRECT D1 LOW TO X6
0EEE	E0F3	ETTR 039		041	DONE	BR IF HZ=0	BR IF FINISHED
0EF0	0206	ETTR 040	RETURN			RTN U MMSK3=0	
0EF2	CDC9	ETTR 041	DONE	046	XFER	BR IF RS BIT0=1	BR IF NOT GATE RD CMLTE(SX041)
0EF4	DDF8	ETTR 042		044	INLK	BR IF RS BIT1=0	BR IF NOT READY(SX041)
0EF6	2B20	ETTR 043	RREQUEST			SET R K=02	SET RDR STATUS REQUEST(RU015)
0EF8	0D04	ETTR 044	INLK			RST RP K=20	RST CMD INLK (RU021)
0EFA	8EFE	ETTR 045		049	SURE	BR	MAKE SURE ADDR SITTING RIGHT
0EC8	3625	ETTR 046	XFER			DO=DO\$K20	SET XFER BIT
0ECA	8EF6	ETTR 047		043	RREQUEST	BR	
0EFC	5BDF	ETTR 048	SETUP			P1=RP1	GET READ 1 INFO(START SHIFT REG)
0EFE	30B9	ETTR 049	SURE			U0=0-K80	
0F00	21A5	ETTR 050				U1=0\$KAO	U=4FA0
0F02	2745	ETTR 051				D1=0\$K40	
0F04	8EEC	ETTR 052		038	D1ADD6	BR	GD MAKE D1=46

 * CROSS REFERENCE FOR CSECT ETTR *

ETTR 022 ETTR 020
 ETTR 036 ETTR 027 ETTR 028
 ETTR 038 ETTR 052
 ETTR 040 ETTR 037

* CROSS REFERENCE FOR CSECT ETTR *

ETTR 041	ETTR 039	
ETTR 043	BMCK 040	ETTR 047
ETTR 044	ETTR 042	
ETTR 046	ETTR 041	
ETTR 048	ETTR 022	
ETTR 049	ETTR 045	

EXFR DESCRIPTIVE TEXT

ENTRY POINTS

<p>XFER ENTRY IS MADE HERE FROM THE ERDR ROUTINE TO SET UP THE ADDRESSES OF THE COLUMN AND ROW IMAGE BUFFERS, AND START THE TRANSFER OF ROW IMAGE TO COLUMN BINARY FORM.</p> <p>BYTECT ENTRY IS MADE HERE FROM THE ERDR ROUTINE AFTER THE ADDRESSES FOR PFR ROW IMAGE AND PFR COLUMN IMAGE BUFFERS HAVE BEEN SET UP. THE TRANSFER OF PFR ROW IMAGE TO PFR COLUMN IMAGE BEGINS HERE.</p> <p>CHKCHK ENTRY IS MADE HERE FROM THE ERDR ROUTINE WHEN A READ-NO-FEED OPERATION IS DETECTED. A CHECK IS MADE FOR READ AND VALIDITY ERRORS. IF NO ERRORS DETECTED, GO TO STORE INDICATOR AND SENSE BYTES IN AUXILIARY STORAGE.</p>	<p>COMBIN ENTRY IS FROM THE ERCX ROUTINE WHEN A VALIDITY CHECK IS DETECTED DURING A PFR OPERATION. UNIT CHECK BIT IS SET, COMMAND CHAINING IS BROKEN, AND THE INDICATOR AND SENSE BYTES ARE STORED IN AUXILIARY STORAGE.</p> <p>WLRBR ENTRY IS FROM THE ECOL ROUTINE. THE WLR INDICATOR BIT HAD BEEN SET IF AN INCORRECT COUNT WAS DETECTED IN THE ECOL ROUTINE. THE ENTRY HERE IS TO TEST FOR SLI AND CHAINING, PRIOR TO STORING THE INDICATOR AND SENSE BYTES IN AUXILIARY STORAGE.</p>
--	---

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		EXFR 001	T			READER XFER OF ROW IMAGE TO COLUMN BINARY FORM	
		EXFR 005	ASEQ	AL07=50			
1050	2455	EXFR 007	XFER			G0=0\$K50	SET G TO 2ND BYTE OF
1052	2515	EXFR 008				G1=0\$K10	COL BUFFER
1054	3593	EXFR 009				G1=G1\$K09	ADDR 5019
1056	2045	EXFR 010				U0=0\$K40	SET U TO ROW IMAGE ADDRESS
1058	21A5	EXFR 011				U1=0\$KAO	40AO
105A	22A3	EXFR 012	BYTECT			V0=0\$K0A	10 GROUPS OF 8 BYTES CNTR
105C	2363	EXFR 013	BITCT			V1=0\$K06	6 BIT PER BYTE CNTR
105E	4852	EXFR 014				RDH I DC,9A	I=C000
1060	4A86	EXFR 015				T=I	CLEAR
1062	4C86	EXFR 016				P=I	WORK
1064	4E86	EXFR 017				H=I	REGISTERS
1066	2613	EXFR 018				DO=0\$K01	SET MASK FOR COL 1
1068	57C0	EXFR 019	RDBYTE			RDB D1 AS,U	READ BYTE FROM IMAGE
106A	C76F	EXFR 020		022	N	BR IF D1 BIT0=1	BITS ARE COMPLIMENT
106C	6965	EXFR 021				I1=I1\$D0	OR BIT IN COL 1
106E	D773	EXFR 022	N	024	NE	BR IF D1 BIT1=1	
1070	6865	EXFR 023				T1=T1\$D0	COL 2
1072	E777	EXFR 024	NE	026	NEX	BR IF D1 BIT2=1	
1074	6D65	EXFR 025				P1=P1\$D0	COL 3
1076	F77B	EXFR 026	NEX	028	NEXT	BR IF D1 BIT3=1	
1078	6F65	EXFR 027				H1=H1\$D0	COL 4
107A	C37F	EXFR 028	NEXT	030	NEXTB	BR IF D1 BIT4=1	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
107C	6E65	EXFR 029				IO=IO\$D0	COL 5
107E	D303	EXFR 030	NEXTB	032	NEXTBT	BR IF D1 BIT5=1	
1080	6A65	EXFR 031				TO=TO\$D0	COL 6
1082	E3C7	EXFR 032	NEXTBT	034	BTNEXT	BR IF D1 BIT6=1	
1084	6C65	EXFR 033				PO=PO\$D0	COL 7
1086	F30B	EXFR 034	BTNEXT	036	BTDCNE	BR IF D1 BIT7=1	
1088	6E65	EXFR 035				HO=HO\$D0	COL 8
108A	6663	EXFR 036	BTDCNE			DO=DO+D0	SHIFT MASK LEFT ONE
108C	23FF	EXFR 037				V1=V1+KFF	BIT COUNT MINUS ONE
108E	C49F	EXFR 038		050	STORE	BR IF Z=0	
1090	F41A	EXFR 040		047	RPFR1	BR IF G0 BIT3=0	BR IF PFR READ
1092	21AB	EXFR 042				U1=U1+K0A	NO- INCREMENT ADR
1094	F498	EXFR 043		045	READM	BR IF AC=0	
1096	201D	EXFR 044				U0=U0+K10	UPDATE TO 60
1098	9068	EXFR 045	READM	019	RDBYTE	BR	
109A	319B	EXFR 047	RPFR1			U1=U1-K09	DECREMENT ADR BY 10
109C	9C68	EXFR 048		019	RDBYTE	BR	
109E	7940	EXFR 050	STORE			STB I1 AS,G+0	STORE COL ONE OR FIVE
10A0	6444	EXFR 051				G=G+2	**POINT ADDR TO EVERY OTHER BYTE
10A2	7B40	EXFR 052				STB T1 AS,G+0	COL TWO OR SIX
10A4	6444	EXFR 053				G=G+2	
10A6	7D40	EXFR 054				STB P1 AS,G+0	COL THREE OR SEVEN
10A8	6444	EXFR 055				G=G+2	
10AA	7F40	EXFR 056				STB H1 AS,G+0	COL FOUR OR EIGHT
10AC	6444	EXFR 057				G=G+2	
10AE	144B	EXFR 058				GO=GO^K04	**INVERT BIT 5 - 2ND 4 BYTES
10B0	D03C	EXFR 059		065	STRDON	BR IF G0 BIT5=0	BR IF DONE 2ND 4 BYTE
10B2	5859	EXFR 060				I1=IO	MOVE
10B4	5AB9	EXFR 061				T1=TO	2ND
10B6	5CD9	EXFR 062				P1=PO	4
10B8	5EF9	EXFR 063				H1=H0	BYTES
10BA	9C9E	EXFR 064		050	STORE	BR	
10BC	22FF	EXFR 065	STRDON			V0=V0+KFF	GROUP CNTR MINUS
10BE	C4CF	EXFR 066		078	HALF	BR IF Z=0	BR IF DONE HALF
10C0	F44A	EXFR 068		075	RPFR2	BR IF G0 BIT3=0	BR IF PFR READ
10C2	313D	EXFR 070				U1=U1-K30	NEXT GROUP OF 8 BYTES
10C4	CC48	EXFR 071		073	HAF1ST	BR IF G0 BIT4=0	BR IF 1ST HALF
10C6	20FD	EXFR 072				U0=U0+KFO	2ND HALF, 50XX
10C8	9C5C	EXFR 073		013	BITCT	BR	GO DO ANOTHER GROUP
10CA	213F	EXFR 075	HAF1ST			U1=U1+K33	NEXT GROUP OF 8 BYTES
10CC	905C	EXFR 076	RPFR2	013	BITCT	BR	GO DO ANOTHER GROUP
10CE	148B	EXFR 078	HALF			GO=GO^K08	**INVERT BIT 4 - 2ND HALF
10D0	C060	EXFR 079		090	DONE	BR IF G0 BIT4=0	BR IF DONE 2ND HALF
10D2	F45A	EXFR 081		087	RPFR3	BR IF G0 BIT3=0	BR IF PFR READ
10D4	211B	EXFR 083				U1=U1+K01	U=40DC
10D6	35AD	EXFR 084				G1=G1-KA0	G=5918
10D8	9C5A	EXFR 085		012	BYTECT	BR	GO DO 2ND HALF
10DA	21A7	EXFR 087	RPFR3			U1=0\$KAA	U=10AA PFR READ
10DC	2507	EXFR 088				G1=0	G=4900
10DE	905A	EXFR 089		012	BYTECT	BR	GO DO 2ND HALF
10E0	F475	EXFR 090	DONE	100	ENDING	BR IF G0 BIT3=1	BR IF NOT PFR READ
10E2	52E2	EXFR 091				RDH V DA,BC	PFR READ,RESTORE DATA ADDRESS
10E4	A062	EXFR 092		ERRQ 087	RPHUCW	BAL	SET D1 TO E8

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
10E6	51A2	EXFR 093				RDB U1 DA,AC	RESTORE CCW ADJUSTED COUNT
10E8	AC44	EXFR 094		ERRQ 092	RDIND	BAL	PU=IND,P1=SENSE
10EA	54B2	EXFR 095				RDH G DA,AE	RESTORE CMD AND FLAGS
10EC	3659	EXFR 096				DO=0-K50	WLR COUNT DO=AF
10EE	2845	EXFR 097				IO=0\$K40	
10F0	29C7	EXFR 098				I1=0	I=4000 POINT TO PFR COL BUFFER
10F2	58DE	EXFR 099		ERDR 164	BRSKIP	BR	
10F4	2466	EXFR 100	ENDING			SET MODE K=36	READER MODE AND ZONE
10F6	E67E	EXFR 104		108	RUNCN	BR IF DO BIT2=0	BR IF NO EXFR
10F8	1625	EXFR 105				DO=DO*-K20	XFER,RST BIT
10FA	24C0	EXFR 106				SET MODE K=00	CPU MODE AND ZONE
10FC	AAA2	EXFR 107		BSWI 066	RESTRH	BR	
10FE	2406	EXFR 108	RUNCN			SET MODE K=30	READER MODE AND ZONE
1100	AC6A	EXFR 109		ERRQ 075	RRDUJW	BAL	SET D=00D8
1102	A044	EXFR 110		ERRQ 092	RDIND	BAL	
1104	5FB2	EXFR 111				RDB H1 DA,AE	RESTORE CMD
1106	F80A	EXFR 112		114	CHKCHK	BR IF H1 BIT7=0	BR IF NOT CMD IMMEDIATE
1108	2643	EXFR 113				DO=0\$K04	CMD IMMEDIATE,SET INDICATOR
110A	FD8E	EXFR 114	CHKCHK	116	NORDCK	BR IF RS BIT3=0	BR IF NO READ CHECK(SX042)
110C	3D15	EXFR 115				P1=P1\$K10	READ CHK,
110E	C992	EXFR 116	NORDCK	118	COMBIN	BR IF RS BIT4=0	BR IF NO VAL CHK(SX043)
1110	3D83	EXFR 117				P1=P1\$K08	VAL CHK,SET INDICATOR IN SENSE
1112	2E23	EXFR 118	COMBIN			HO=0\$K02	UNIT CHECK
1114	34E5	EXFR 119		125	BRKCC N	GO=GO\$K80	SET SECONDARY BIT
1116	A923	EXFR 120				BR IF P1=NZ	BR IF NO SENSE BIT
1118	2E07	EXFR 121				HO=0	NO UNIT CHK
111A	F822	EXFR 122	WLRBR	126	CCBR	BR IF P0 BIT7=0	BR IF NO WLR
111C	E523	EXFR 123		126	CCBR	BR IF G1 BIT2=1	WLR, BR IF SLI
111E	3445	EXFR 124				GO=GO\$K40	WLR
1120	1545	EXFR 125	BRKCC 0			G1=G1*-K40	RESET CC BIT
1122	8564	EXFR 126	CCBR	ERRQ 102	STIND	BAL	STORE IND., AND SENSE
1124	D529	EXFR 127		129	CC	BR IF G1 BIT1=1	BR IF CC
1126	3E83	EXFR 128				HO=HO\$K08	NO CC,SET CHAN END BIT
1128	D22E	EXFR 129	CC	132	NOTIME	BR IF DO BIT5=0	BR IF NOT CMD IMMEDIATE
112A	56F2	EXFR 130				RDH D DA,BE	CMD IMMEDIATE, RESTORE DO BIT 7
112C	6D58	EXFR 131		DPTR 080	IMMEDI	BR	
112E	51A2	EXFR 132	NOTIME			RDB U1 DA,AC	RESTORE RESIDUE COUNT
1130	2CC7	EXFR 133				UO=0	HIGH COUNT EQUALS ZERO
1132	D83A	EXFR 134		138	HICTO	BR IF P0 BIT5=0	BR IF NO EXCESS COUNT STORED
1134	52F2	EXFR 135				RDH V DA,BE	READ EXCESS COUNT
1136	613B	EXFR 136				U1C=U1+V1	ADD BACK THE
1138	6C2D	EXFR 137				UOC=UO+VO+C	EXCESS COUNT
113A	C844	EXFR 138	HICTO	143	NOIPL	BR IF P0 BIT4=0	BR IF FEED
113C	3E43	EXFR 139				HO=HO\$K04	NO FEED, SET DE
113E	C3C4	EXFR 140		143	NOIPL	BR IF BA BIT4=0	BR IF NOT IPL
1140	2B20	EXFR 141				SET R K=02	IPL,SET RDR REQUEST(RU015)
1142	9D54	EXFR 142	GOSTUC	DPTQ 020	STINST	BR	STORE UCW, GO TO BSWI
1144	C3C3	EXFR 143	NOIPL	142	GOSTUC	BR IF BA BIT4=1	NO INTERRUPT FOR IPL
1146	8D44	EXFR 144		DPTR 066	CCBR	BR	INT, OR GO CC,STORE UCW, GO BSWI
		EXFR 145	AEND				

 * CROSS REFERENCE FOR CSECT EXFR *

EXFR 007	ERDR 181	ERDR 183	ERDR 185
EXFR 012	ERDR 153	EXFR 085	EXFR 089
EXFR 013	EXFR 073	EXFR 076	
EXFR 019	EXFR 045	EXFR 048	
EXFR 022	EXFR 020		
EXFR 024	EXFR 022		
EXFR 026	EXFR 024		
EXFR 028	EXFR 026		
EXFR 030	EXFR 028		
EXFR 032	EXFR 030		
EXFR 034	EXFR 032		
EXFR 036	EXFR 034		
EXFR 045	EXFR 043		
EXFR 047	EXFR 040		
EXFR 050	EXFR 038	EXFR 064	
EXFR 065	EXFR 059		
EXFR 073	EXFR 071		
EXFR 075	EXFR 068		
EXFR 078	EXFR 066		
EXFR 087	EXFR 081		
EXFR 090	EXFR 079		
EXFR 100	EXFR 050		
EXFR 108	EXFR 104		
EXFR 114	ERDR 119	EXFR 112	
EXFR 116	EXFR 114		
EXFR 118	ERCX 049	EXFR 116	
EXFR 122	ECOL 091		
EXFR 125	EXFR 120		
EXFR 126	EXFR 122	EXFR 123	
EXFR 129	EXFR 127		
EXFR 132	EXFR 129		
EXFR 138	EXFR 134		
EXFR 142	EXFR 143		
EXFR 143	EXFR 138	EXFR 140	

FILE DESCRIPTIVE TEXT

2311 UCW

```
AUX 0X8E-0X8F          ZONE 1 T-REG  ZONE 1 I-REG  AUX 0X9E-0X99
*****
*      *      *      *      *      *
* STATUS-I/O ADR * FLAGS AND OP * DATA ADDRESS * COUNT FIELD * NEXT CCW ADDR *
*      * (FFI)  (FOP)*      *      *      *
*****
```

STATUS-I/O ADR

FLAGS AND OP

0=ACTIVE
 1=CAME FROM START I/O
 2=RETURN TO ICYCLES
 3=PERFORMED HALT I/O
 4-15=FILE I/O ADDRESS

0=CHAIN DATA
 1=CC OR CD AND NOT UNIT CHECK
 2=CC OR CD AND NOT UNIT CHECK
 3=SKIP
 4=PCI
 5=SET INTERRUPT LATCH (STATUS
 IN INTERRUPT BUFFER).
 6=ANY INTERRUPT CONDITON
 7=SLI

ENTRY POINTS

ENTRY THIS IS THE NORMAL ENTRY FOR A PROGRAMMED INSTRUCTION. THIS ROUTINE DETERMINES- START I/O, TEST I/O, TEST CHANNEL, OR HALT I/O AND BRANCHES FOR REQUIRED PROCESS. THE START I/O SEQUENCE TESTS THE AVAIL- ABILITY OF THE FILE, AND DETERMINES THE FILE READY CONDITION. THE FILE CONTROLS ARE TESTED FOR OPERATION.

TSTCC THIS ENTRY ALLOWS THE USE OF THE CYCLIC-CODE AND OPERATIONAL TEST ROUTINE AFTER CONTROL AND SENSE OPERATIONS BEFORE PROCEEDING WITH OPERATING COMMANDS.

SNSSTA THIS IS A BAL ENTRY TO SET AN EQUIPMENT CHECK AND SELECTED STATUS INDICATIONS IN THE SENSE BYTES.

SETDA THIS IS A BAL ENTRY USED TO SET THE DEVICE ADDRESS FROM AN UNSELECTED GATED ATTENTION. AFTER DEVELOPING THE DEVICE ADDRESS, THE INDICATED MODULE IS SELECTED.

RSTATT THIS IS A BAL ENTRY USED TO RESET THE FILE GATED ATTENTION. THE READ-GATE IS RAISED MOMENTARILY FOR THE RESET AND THEN THE MODULE IS DESELECTED.

2311 AUX MODULE 0 USAGE

```
*****
* ADDRESS *
* 04 AND 05 - DEVICE ADDRESS FOR *
* CONTINGENT CONNECTION*
* 06 AND 07 - INTERRUPT BUFFER *
* 14 AND 15 - ERROR STATUS (CE) *
* 16 AND 17 - ERROR NEXT CCW ADDR *
* 8E AND 8F - STATUS AND CHANNEL *
* DEVICE ADDRESS. *
* 98 AND 99 - NEXT CCW ADDRESS *
* 9A AND 9B - SENSE BYTES 0 AND 1 *
* OR PREVIOUS OP AND *
* MASK. *
* 9C AND 9D - SENSE BYTES 0 AND 1 *
* OR FILE ADDRESS *
* F5 - CONTROL UNIT ADDRESS*
*****
```

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		FILE 001	T			2311 FEATURE COMMON ENTRY FOR START IO,TEST IO,HALT IO	
		FILE 002	*			*****	
		FILE 003	*			*	**
		FILE 004	*			FILE TIO - SIO - HID COMMON	**
		FILE 005	*			*	**
		FILE 006	*			*****	
		FILE 007	ASEQ	AL07=7A			
047A	7CC2	FILE 008	ENTRY			STH P DA,88	STORE CPU P REG
047C	2480	FILE 009				SET MODE K=08	SET FILE MODE & CPU ZONE
047E	C628	FILE 010		031	AVAIL	BR IF DO BIT0=0	BR IF CHANNEL AVAILABLE
0480	E011	FILE 011		019	HALTIO	BR IF GO BIT6=1	BR IF HALT IO
0482	F01E	FILE 012		026	SETCC2	BR IF GO BIT7=0	BR IF NOT TEST IO
0484	D99E	FILE 013		026	SETCC2	BR IF FF15=0	BR IF INTURRUPT LATCH NOT ON
0486	6711	FILE 014				D1=D10U1	COMPARE IOA
0488	C49E	FILE 015		026	SETCC2	BR IF ZNZ	BR IF IOA NOT SAME
048A	7032	FILE 016				STH U DA,8E	SAVE IOA WITH RESET STATS
048C	8420	FILE 017		FINT 052	SETCSW	BAL	SET REGS TO USE CPU CSW STORE
048E	A024	FILE 018		CCOM 127	CSWADR	BR	
0490	5DBF	FILE 019	HALTIO			T1=FF0	GET CCW FLAGS
0492	1BC5	FILE 020				T1=T1*-KC0	STRIP CC AND DC FLAGS
0494	D9A1	FILE 021		027	SETHIO	BR IF FF15=1	BR IF INTURRUPT LATCH IS ON
0496	3210	FILE 022				SET MMSK K=81	SET PRIORITY
0498	2D04	FILE 023				SET FIA K=20	SET CCW COUNT 000
049A	4DBF	FILE 024				FF0=T1	RESTORE CCW FLAGS WITHOUT CHAIN
049C	1210	FILE 025				RST MMSK K=81	RESET PRIORITY
049E	AC74	FILE 026	SETCC2	CCOM 161	CC2B	BR	BR TO SET CC2
04A0	F5BB	FILE 027	SETHIO	040	SETCUB	BR IF DAS13=1	BR IFERASING
04A2	2E07	FILE 028				H0=0	ZERO DEVICE STATUS
04A4	D9BE	FILE 029		042	HTS	BR IF FF15=0	BR IF NOT INTURRUPT LATCH
04A6	B362	FILE 030		CCOM 154	CC0B	BR	BR TO SET CC0
04A8	3065	FILE 031	AVAIL			U0=U0\$K60	SET FROM SIO&RTN TO I CYC STATS
04AA	7032	FILE 032				STH U DA,8E	STORE STATS AND I/O ADDRESS
04AC	E021	FILE 033		027	SETHIO	BR IF GO BIT6=1	BR IF HALT I/O
04AE	F5BB	FILE 034		040	SETCUB	BR IF DAS13=1	BR IF ERASING
04B0	E1C1	FILE 035		043	TSTCUE	BR IF DAS16=1	BR IF CUE
04B2	F054	FILE 036		053	SIO	BR IF GO BIT7=0	BR IF SIO
04B4	DBFD	FILE 037		073	GETDS	BR IF FOP5=1	BR IF CONTIN CONNECT CLEAR
04B6	8702	FILE 038		250	TSTADR	BAL	TEST FOR CONTINGENT CONNECTION
04B8	DBFC	FILE 039		073	GETDS	BR IF FOP5=0	BR IF NOT CONTIN CONNECT
04BA	2E55	FILE 040	SETCUB			H0=0\$K50	SET CUB STATUS
04BC	2D02	FILE 041				SET FIA K=10	SET CUB LATCH
04BE	854E	FILE 042	HTS	114	SIOTIO	BR	
04C0	DBCA	FILE 043	TSTCUE	048	CUE	BR IF FOP5=0	BR IF NO CONTIN CONNECT
04C2	8702	FILE 044		250	TSTADR	BAL	TST FOR CONTIN CONNECT
04C4	DBBB	FILE 045		040	SETCUB	BR IF FOP5=1	BR IF CONTIN CONNECT
04C6	2E43	FILE 046				H0=0\$K04	SET CONTINGENT
04C8	4FEF	FILE 047				FOP=H0	CONNECTION CLEAR INDICATOR
04CA	2E25	FILE 048	CUE			H0=0\$K20	SET CUE FOR STATUS
04CC	F051	FILE 049		051	RSTCUE	BR IF GO BIT7=1	BR IF TEST I/O
04CE	2E35	FILE 050				H0=0\$K30	SET CUE & BUSY STATUS
04D0	0D02	FILE 051	RSTCUE			RST FIA K=10	RESET CUB LATCH
04D2	854E	FILE 052		114	SIOTIO	BR	
04D4	7242	FILE 053	SIO			STH V DA,98	SAVE CCW ADDRESS

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
04D6	5AF2	FILE 054				RDH T CA,BE	GET STOR PROTECT KEY
04D8	42AF	FILE 055				STPO=TO	SET PROTECT KEY
04DA	5A30	FILE 056				RDH T V+0	GET CCW COMMAND
04DC	0A48	FILE 057				Z=TO=K04	MASK FOR SENSE OP
04DE	C4E4	FILE 058		061	NOTSNS	BR IF ZNZ	BR IF NOT SENSE
04E0	AFC6	FILE 059		296	MODSEL	BAL	SELECT MODULE FOR SENSE
04E2	8C92	FILE 060		222	EXCUTE	BR	
04E4	DBEB	FILE 061	NOTSNS	064	TSTNOP	BR IF FOP5=1	BR IF CONTIN CONECT CLEAR
04E6	8702	FILE 062		250	TSTADR	BAL	TST FOR CONTINGENT CONNECTION
04E8	DBBB	FILE 063		040	SETCUB	BR IF FOP5=1	BR IF CONTIN CONECT
04EA	0A3B	FILE 064	TSTNOP			Z=TO=K03	MASK FOR NO OP
04EC	C4FD	FILE 065		073	GETDS	BR IF Z=0	BR IF NO OP
04EE	0A1D	FILE 066				Z=TO=K10	MASK FOR HI BIT OF RESTORE
04F0	E0F6	FILE 067		070	TSTIOA	BR IF HZNZ	BR IF NOT RESTORE
04F2	0A7B	FILE 068				Z=TO=K07	MASK FOR LO BITS OF RESTORE
04F4	F0FD	FILE 069		073	GETDS	BR IF LZ=0	BR IF RESTORE
04F6	0133	FILE 070	TSTIOA			Z=U1*-K03	MASK FOR VALID DEVICE ADDR
04F8	F0FD	FILE 071		073	GETDS	BR IF LZ=0	BR IF DA VALID
04FA	850C	FILE 072		081	SETIVR	BR	
04FC	5ECF	FILE 073	GETDS			PO=DS	GET DISK STATUS
04FE	C487	FILE 074		078	BALMS	BR IF Z=0	BR IF NO UNSELECTED STATUS
0500	2C43	FILE 075				PO=0\$K04	SET UNSELECTED STATUS BIT
0502	9490	FILE 076	STRSEN	265	SNSSTB	BAL	STORE SENSE
0504	854A	FILE 077		112	STR04	BR	
0506	AFC6	FILE 078	BALMS	296	MODSEL	BAL	BAL TO MOD SELECT
0508	EE93	FILE 079		084	UNSAFE	BR IF DS2=1	BR IF UNSAFE
050A	DE97	FILE 080		086	TSTDST	BR IF DS1=1	BR IF ON LINE
050C	8114	FILE 081	SETIVR	336	ZRCSEN	BAL	ZERO SENSE BYTES. INVALID 2311
050E	2A45	FILE 082				TO=0\$K40	SET INT REQD
0510	8548	FILE 083		111	CONCON	BR	BR TO SET CONTINGENT CON
0512	2C85	FILE 084	UNSAFE			PO=0\$K80	SET UNSAFE FOR SENSE
0514	85C2	FILE 085		076	STRSEN	BR	
0516	2E15	FILE 086	TSTDST			HO=0\$K10	PRESET BUSY STATUS
0518	FAA9	FILE 087		095	SKINC	BR IF DS7=1	BR IF SEEK INCOMPLETE
051A	CEBC	FILE 088		105	BRANCH	BR IF DS0=0	BR IF NOT READY
051C	E5B5	FILE 089		101	ATNRST	BR IF DAS12=1	BR IF SELECTED GATED ATTENTION
051E	F023	FILE 090		092	TIO	BR IF GO BIT7=1	BR IF TIO
0520	8BE8	FILE 091		137	CCTEST	BR	BR TO CC TEST OF SID ROUTINE
0522	49AF	FILE 092	TIO			MS=TO	DESELECT FILE
0524	5CC2	FILE 093				RDH P DA,B8	RESTORE CPU P REG
0526	B362	FILE 094		CCOM 154	CCOB	BR	SET CCO
0528	8114	FILE 095	SKINC	336	ZRCSEN	BAL	
052A	CA3B	FILE 096				Z=TO=K03	MASK FOR LO BITS OF RECAL
052C	F0C4	FILE 097		109	RSTATN	BR IF LZNZ	BR IF NOT RECAL
052E	0A1D	FILE 098				Z=TO=K10	MASK FOR HI BITS RECAL
0530	E0C4	FILE 099		109	RSTATN	BR IF HZNZ	BR IF NOT RECALIBRATE
0532	8C92	FILE 100		222	EXCUTE	BR	
0534	8224	FILE 101	ATNRST	319	RSTATT	BAL	BAL TO RESET GATED ATTN
0536	2E43	FILE 102				HO=0\$K04	SET DE BIT
0538	F03D	FILE 103		105	BRANCH	BR IF GO BIT7=1	BR IF TEST I/O
053A	3E15	FILE 104				HO=HO\$K10	OR IN BUSY STATUS FOR SID
053C	8230	FILE 105	BRANCH	325	RSTMOD	BAL	
053E	F051	FILE 106		115	MODE	BR IF GO BIT7=1	BR IF TEST IO

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0540	8114	FILE 107		336	ZROSEN	BAL	BAL TO ZERO SENSE BYTES
0542	855A	FILE 108		120	TSTPCI	BR	
0544	8224	FILE 109	RSTATN	319	RSTATT	BAL	BAL TO RESET GATED ATTN
0546	2A13	FILE 110				TO=0\$K01	SET SEEK CHECK SENSE 0-7
0548	9496	FILE 111	CONCON	268	SNSSTC	BAL	STORE SENSE
054A	2843	FILE 112	STR04			T1=0\$K04	
054C	70A0	FILE 113				STH U AS,T+0	
054E	F058	FILE 114	SIOTIO	119	TSTHIO	BR IF GO BIT7=0	BR IF NOT TIO
0550	2400	FILE 115	MODE			SET MODE K=00	SET CPU MODE AND ZONE
0552	5CC2	FILE 116				RDH P DA,B8	RESTORE CPU P REG
0554	2FC7	FILE 117				H1=0	
0556	A01E	FILE 118		CCOM 123	STATOS	BR	
0558	E063	FILE 119	TSTHIO	124	ZERCHN	BR IF GO BIT6=1	BR IF HIO
055A	234B	FILE 120	TSTPCI			V1=V1+K04	ADD 4 TO CCW ADDR
055C	5630	FILE 121				RDH D V+0	
055E	2F85	FILE 122				H1=0\$K80	SET PCI BIT FOR CHNL STATUS
0560	C365	FILE 123		125	ZERT	BR IF D1 BIT4=1	BR IF PCI IN CURRENT CCW
0562	2FC7	FILE 124	ZERCHN			H1=0	ZERO CHANNEL STATUS
0564	2A07	FILE 125	ZERT			TO=0	ZERO HI ADDR BYTE
0566	C3E0	FILE 126		129	BAON	BR IF BA4=1	BR IF IPL
0568	2400	FILE 127				SET MODE K=00	SET CPU MODE AND ZONE
056A	A030	FILE 128		CCOM 135	STATUS	BR	BR TO STORE SRATUS AND SET CC1
056C	50C7	FILE 129	BAON			STOP	
		FILE 130	AEND				
		FILE 131	*				
		FILE 132	*				**
		FILE 133	*			CYCLIC CODE HARDWARE TEST	**
		FILE 134	*				**
		FILE 135	*				**
		FILE 136	ASEQ	AL07=68			
08E8	8114	FILE 137	CCTEST	336	ZROSEN	BAL	CLEAR THE SENSE BUFFER
08EA	3E00	FILE 138				SET FIB K=80	SET NTO LATCH
08EC	2490	FILE 139				SET MODE K=09	SET FILE MODE AND ZONE
08EE	0E08	FILE 140				RST FIB K=40	INITIAL RESET
08F0	2302	FILE 141				SET DIAC K=10	SET DIAGNOSTIC MODE
08F2	2304	FILE 142				SET DIAC K=20	SET DIAG INDEX
08F4	3110	FILE 143				SET DIAB K=81	SET DIAG ADDR 0
08F6	3D29	FILE 144				P1=0-K20	SET CONSTANT DF
08F8	4FDF	FILE 145				FOP=P1	LOAD OP REG
08FA	43DF	FILE 146				FEBO=P1	LOAD WRITE BUFFER
08FC	B344	FILE 147		243	PEDS	BAL	TEST FOR CORRECT BITS IN WR BUFF
08FE	5FCF	FILE 148				PO=FOP	GET OP REG BITS
0C00	1CFF	FILE 149				PO=PO^KFF	MASK FOR CORRECT OP REG BITS
0C02	C4FE	FILE 150		212	ALUCHK	BR IF ZNZ	BR IF OP REG ERROR
0C04	4FCF	FILE 151				FOP=PO	RESET OP REG
0C06	5FCF	FILE 152				PO=FOP	GET OP REG BITS
0C08	C4FE	FILE 153		212	ALUCHK	BR IF ZNZ	BR IF OP REG NOT RESET
0C0A	8CCA	FILE 154		233	SHIFT8	BAL	ADVANCE BIT RING
0C0C	5BCF	FILE 155				PO=SDI	GET READ BUFF BITS
0C0E	B346	FILE 156		244	XOR	BAL	TST XFER OF WR TO RD BUFF
0C10	5ECF	FILE 157				PO=DS	
0C12	C4FE	FILE 158		212	ALUCHK	BR IF ZNZ	BR IF WR BUFF NOT RESET
0C14	2310	FILE 159				SET DIAC K=01	ADV TO ZONE 1

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0C16	2390	FILE 160				SET DIAC K=09	ADV TO ZONE 2&SET SEP DATA
0C18	2D13	FILE 161				P1=0\$K01	SET WRITE BIT
0C1A	4FDF	FILE 162				FOP=P1	LOAD OP REG
0C1C	2D25	FILE 163				P1=0\$K20	
0C1E	43DF	FILE 164				FEBD=P1	LOAD WR BUFF
0C20	3110	FILE 165				SET DIAB K=81	SET READ & COMPARE GATES
0C22	8CD6	FILE 166		239	SHIFT2	BAL	ADV BIT RING
0C24	2D35	FILE 167				P1=0\$K30	SET SCH EQ ID
0C26	2D1B	FILE 168				P1=P1+K01	''
0C28	4FDF	FILE 169				FOP=P1	LOAD OP REG
0C2A	8CCE	FILE 170		235	SHIFT6	BAL	ADV BIT RING
0C2C	5BCF	FILE 171				PO=SDI	GET READ BUFF BITD
0C2E	2D85	FILE 172				P1=0\$K80	
0C30	3D23	FILE 173				P1=P1\$K02	
0C32	B346	FILE 174		244	XOR	BAL	TST FOR CORRECT WR BUFF SHUFT
0C34	3D73	FILE 175				P1=P1\$K07	
0C36	B342	FILE 176		242	SET3	BAL	TST FOR CORRECT BITS IN CC REG
0C38	3114	FILE 177				SET DIAB K=A1	SET READ & COMPARE GATES
0C3A	8CD2	FILE 178		237	SHIFT4	BAL	ADV BIT RING
0C3C	2DA3	FILE 179				P1=0\$K0A	
0C3E	B342	FILE 180		242	SET3	BAL	TST FOR CORRECT CC DATA
0C40	1D00	FILE 181				RST FIA K=80	RST HI LO CC ERR&TRAP LATCHES
0C42	CAFF	FILE 182		212	ALUCHK	BR IF DS4=1	BR IF UNEQUAL NOT RESET
0C44	2380	FILE 183				SET DIAC K=08	SET SEPERATED DATA
0C46	3114	FILE 184				SET DIAB K=A1	SET READ & COMPARE GATES
0C48	8CD2	FILE 185		237	SHIFT4	BAL	ADV BIT RING
0C4A	2D45	FILE 186				P1=0\$K40	
0C4C	3DE3	FILE 187				P1=P1\$K0E	
0C4E	B342	FILE 188		242	SET3	BAL	TEST FOR CORRECT CC DATA
0C50	1D00	FILE 189				RST FIA K=80	RST HI LO CC ERR&TRAP LATCHES
0C52	2390	FILE 190				SET DIAC K=09	ADV TO ZONE 3&SET SEP DATA BIT
0C54	3114	FILE 191				SET DIAB K=A1	SET READ & COMPARE GATES
0C56	2308	FILE 192				SET DIAC K=40	ADV BIT RING
0C58	2D75	FILE 193				P1=0\$K70	
0C5A	3DA3	FILE 194				P1=P1\$K0A	
0C5C	B342	FILE 195		242	SET3	BAL	TEST FOR CORRECT CC DATA
0C5E	1D00	FILE 196				RST FIA K=80	RST HI LO CC ERR&TRAP LATCHES
0C60	3114	FILE 197				SET DIAB K=A1	SET READ & COMPARE GATES
0C62	2380	FILE 198				SET DIAC K=08	SET SEP DATA BIT
0C64	2308	FILE 199				SET DIAC K=40	ADV BIT RING
0C66	2D35	FILE 200				P1=0\$K30	
0C68	3DA3	FILE 201				P1=P1\$K0A	
0C6A	B342	FILE 202		242	SET3	BAL	TEST FOR CORRECT CC DATA
0C6C	1D00	FILE 203				RST FIA K=80	RST HI LO CC ERR&TRAP LATCHES
0C6E	3114	FILE 204				SET DIAB K=A1	SET READ & COMPARE GATES
0C70	2308	FILE 205				SET DIAC K=40	ADV BIT RING
0C72	2D55	FILE 206				P1=0\$K50	
0C74	3D83	FILE 207				P1=P1\$K0B	
0C76	B342	FILE 208		242	SET3	BAL	TST FOR CORRECT CC DATA
0C78	1D00	FILE 209				RST FIA K=80	RST HI LO CC ERR&TRAP LATCHES
0C7A	2480	FILE 210				SET MODE K=08	SET FILE MODE CPU ZONE
0C7C	8C8C	FILE 211		219	CCNOER	BR	END OF TEST - RESULTS OK
0C7E	2100	FILE 212	ALUCHK			SET DIAB K=00	RST DIAG ADDR LATCHES

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0C80	1E00	FILE 213				RST FIB K=80	CHAIN END RESET
0C82	0E08	FILE 214				RST FIB K=40	INITIAL RST
0C84	1210	FILE 215				RST MMSK K=81	
0C86	2480	FILE 216				SET MODE K=08	SET FILE MODE-CPU ZONE
0C88	3C83	FILE 217				PO=PO\$K08	SET ALU CHECK SENSE BIT
0C8A	8502	FILE 218		076	STRSEN	BR	
0C8C	2100	FILE 219	CCNDR			SET DIAB K=00	RESET DAAG ADDR LATCHES
0C8E	3E00	FILE 220				SET FIB K=80	SET NTO LATCH
0C90	8C92	FILE 221		222	EXCUTE	BR	
0C92	0EC8	FILE 222	EXCUTE			RST FIB K=40	INITIAL RST
0C94	0E04	FILE 223				RST FIB K=20	CLOLD START
0C96	1210	FILE 224				RST MMSK K=81	
0C98	2D43	FILE 225				P1=0\$K04	SET HD SELECT BIT
0C9A	4EDF	FILE 226				FBO=P1	HD SELECT TO BUSS
0C9C	5CC2	FILE 227				RDH P DA, B8	RESTORE CPU P REG
0C9E	2AC7	FILE 228				TO=0	
0CA0	2490	FILE 229				SET MODE K=09	SET FILE MODE AND ZONE
0CA2	5E42	FILE 230				RDH H DA, 98	RD OUT CCW NA
0CA4	863A	FILE 231		FILT 024	GETOP	BR	BR TO SIO, TRAP COMMON WORD
		FILE 232	AEND				
0CCA	2308	FILE 233	SHIFT8			SET DIAC K=40	
0CCC	2308	FILE 234	SHIFT7			SET DIAC K=40	
0CCE	2308	FILE 235	SHIFT6			SET DIAC K=40	
0CD0	2308	FILE 236	SHIFT5			SET DIAC K=40	
0CD2	2308	FILE 237	SHIFT4			SET DIAC K=40	
0CD4	2308	FILE 238	SHIFT3			SET DIAC K=40	
0CD6	2308	FILE 239	SHIFT2			SET DIAC K=40	
0CD8	2308	FILE 240	SHIFT1			SET DIAC K=40	
0CDA	128E	FILE 241				RTN	
3342	2112	FILE 242	SET3			SET DIAB K=11	SET DIAG ADDR 3
3344	5ECF	FILE 243	PEDS			PO=DS	
3346	6CD1	FILE 244	XOR			PO=PO\$P1	MASK FOR CORRECT BITS
3348	C4CD	FILE 245		247	RETURN	BR IF Z=0	BR IF NO ERROR
334A	8C7E	FILE 246		212	ALUCHK	BR	
334C	128E	FILE 247	RETURN			RTN	
		FILE 248	*			*****	
		FILE 249	*			*****	
						CONTINGENT CONNECTION TEST - SUBROUTINE **	
0702	5C52	FILE 250	TSTADR			RDH P DA, 9A	GET SNS BYTES 0&1
0704	F917	FILE 251		260	PREST	BR IF P1 BIT7=1	BR IF NONSENSE STORED
0706	EC16	FILE 252		260	PREST	BR IF PO BIT2=0	BR IF NO UNSENSED SENSE
0708	2C07	FILE 253				PO=0	SET AUX
070A	2D43	FILE 254				P1=0\$K04	ADDR 0004
070C	5EC0	FILE 255				RDH H AS, P+0	GET I/O ADDR
070E	4FCF	FILE 256				FOP=PO	ZERO OP REG
0710	6F11	FILE 257				H1=HI\$U1	COMPARE I/O ADDRESSES
0712	C497	FILE 258		260	PREST	BR IF Z=0	BR IF COMPARE EQUAL
0714	4FDF	FILE 259				FOP=P1	SET OP REG INDICATOR
0716	5CC2	FILE 260	PREST			RDH P DA, B8	RESTORE CPU P REG
0718	128E	FILE 261				RTN	
		FILE 262	*			*****	
		FILE 263	*			*****	
						SET SENSE BYTES - SUBROUTINE **	
148E	2C15	FILE 264	SNSSTA			PO=0\$K10	
1490	2DC7	FILE 265	SNSSTB			P1=0	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1492	7C62	FILE 266				STH P DA,9C	STORE SENSE BYTES 2,3
1494	2A15	FILE 267				T0=0\$K10	
1496	3A25	FILE 268	SNSSTC			T0=T0\$K20	SET CONTINGENT CONN IND.SENSE0-2
1498	2B07	FILE 269				T1=0	
149A	7A52	FILE 270				STH T DA,9A	STORE SENSE BYTES 0,1
149C	2AC7	FILE 271				T0=0	
149E	2E23	FILE 272				H0=0\$K02	SET UNIT CHK DEVICE STATUS
14A0	49BF	FILE 273				MS=T1	DESELECT MOD
14A2	128E	FILE 274				RTN	
		FILE 275	*			*****	
		FILE 276	*			*	**
		FILE 277	*			SET DEVICE ADDRESS	- SUBROUTINE **
		FILE 278	*			FOR GATED ATTENTION	**
		FILE 279	*			*****	
2FEA	2D95	FILE 280	SETDA			P1=0\$K90	SET DEVICE ADDR 0
2FEC	CE08	FILE 281				RST FIB K=40	GIVE INITL RESET TO RST CHN END
2FEE	2100	FILE 282				SET DIAB K=00	
2FF0	CCD1	FILE 283		301	MOD0	BR IF FGA0=1	BR IF MOD 0 GATED ATT
2FF2	2D1B	FILE 284				P1=P1+K01	
2FF4	DCD9	FILE 285		305	MOD1	BR IF FGA1=1	BR IF MOD 1 GATED ATTN
2FF6	2D1B	FILE 286				P1=P1+K01	
2FF8	ECD5	FILE 287		303	MOD2	BR IF FGA2=1	BR IF MOD 2 GATED ATTN
2FFA	2D1B	FILE 288				P1=P1+K01	
2FFC	FCDD	FILE 289		307	MOD3	BR IF FGA3=1	BR IF MOD 3 GATED ATTN
2FFE	128E	FILE 290				RTN	
		FILE 291	*			*****	
		FILE 292	*			*	**
		FILE 293	*			MODULE SELECT	- SUBROUTINE **
		FILE 294	*			*	**
		FILE 295	*			*****	
2FC6	2EC7	FILE 296	MODSEL			H0=0	
2FC8	5C32	FILE 297				RDH P DA,8E	GET I/O ADDR
2FCA	0E08	FILE 298				RST FIB K=40	GIVE INITL RESET TO RST CHN END
2FCC	C965	FILE 299		311	ENDSEL	BR IF P1 BIT4=1	BR IF DRIVE IS 8 OR HIGHER
2FCE	D965	FILE 300		311	ENDSEL	BR IF P1 BIT5=1	BR IF DRIVE IS 4 OR HIGHER
2FD0	2C85	FILE 301	MOD0			PO=0\$K80	
2FD2	E956	FILE 302		304	TEST2	BR IF P1 BIT6=0	
2FD4	2C25	FILE 303	MOD2			PO=0\$K20	
2FD6	F95E	FILE 304	TEST2	308	SELECT	BR IF P1 BIT7=0	
2FD8	2C45	FILE 305	MOD1			PO=0\$K40	
2FDA	E95E	FILE 306		308	SELECT	BR IF P1 BIT6=0	
2FDC	2C15	FILE 307	MOD3			PO=0\$K10	
2FDE	2E07	FILE 308	SELECT			H0=0	
2FE0	4BEF	FILE 309				TGRO=H0	RESET TAG REG
2FE2	49CF	FILE 310				MS=PO	MODULE SELECT
2FE4	4EEF	FILE 311	ENDSEL			FBO=H0	RESET FILE BUS OUT
2FE6	4DEF	FILE 312				FFO=H0	RESET CCW FILE FLAGS
2FE8	128E	FILE 313				RTN	
		FILE 314	*			*****	
		FILE 315	*			*	**
		FILE 316	*			RESET GATED ATTENTION	- SUBROUTINE **
		FILE 317	*			*	**
		FILE 318	*			*****	

ALDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0224	2CG7	FILE 319	RSTATT			PO=0	RESET GATED ATTENTION SUBR
0226	4BCF	FILE 320				TGRO=PO	RESET CONTROL
0228	2D45	FILE 321				P1=0\$K40	
022A	4EDF	FILE 322				FBO=P1	FILE BUS OUT=1BIT
022C	2D15	FILE 323				P1=0\$K10	
022E	4BDF	FILE 324				TGRO=P1	RAISE CONTROL
0230	2DC7	FILE 325	RSTMOD			P1=0	
0232	4BDF	FILE 326				TGRO=P1	RESET CONTROL
0234	4EDF	FILE 327				FBO=P1	RESET FILE BUS
0236	4FDF	FILE 328				FOP=P1	RESET FILE OP REG
0238	49DF	FILE 329				MS=P1	DESELECT MOD
023A	128E	FILE 330				RTN	
		FILE 331	*			*****	
		FILE 332	*			*	
		FILE 333	*			ZERO SENSE BYTES	- SUBROUTINE
		FILE 334	*			*	**
		FILE 335	*			*****	
0114	2CG7	FILE 336	ZROSEN			PO=0	
0116	2D13	FILE 337				P1=0\$K01	
0118	7C52	FILE 338				STH P DA,9A	ZERO SENSE BYTES 0,1
011A	2DC7	FILE 339				P1=0	
011C	7C62	FILE 340				STH P DA,9C	ZERO SENSE BYTES 2,3
011E	128E	FILE 341				RTN	

 * CROSS REFERENCE FOR CSECT FILE *

FILE 008	DCLA 113				
FILE 019	FILE 011				
FILE 026	FILE 012	FILE 013	FILE 015		
FILE 027	FILE 021	FILE 033			
FILE 031	FILE 010				
FILE 040	FILE 027	FILE 034	FILE 045	FILE 063	
FILE 042	FILE 029				
FILE 043	FILE 035				
FILE 048	FILE 043				
FILE 051	FILE 049				
FILE 053	FILE 036				
FILE 061	FILE 058				
FILE 064	FILE 061				
FILE 070	FILE 067				
FILE 073	FILE 037	FILE 039	FILE 065	FILE 069	FILE 071
FILE 076	FILE 085	FILE 218			
FILE 078	FILE 074				
FILE 081	FILE 072				
FILE 084	FILE 079				
FILE 086	FILE 080				
FILE 092	FILE 090				
FILE 095	FILE 087				
FILE 101	FILE 089				
FILE 105	FILE 088	FILE 103			
FILE 109	FILE 097	FILE 099			
FILE 111	FILE 083				
FILE 112	FILE 077				

* CROSS REFERENCE FOR CSECT FILE *

FILE 114	FILE 042	FILE 052							
FILE 115	FILE 106								
FILE 119	FILE 114								
FILE 120	FILE 108								
FILE 124	FILE 119								
FILE 125	FILE 123								
FILE 129	FILE 126								
FILE 137	FILE 091								
FILE 212	FILE 150	FILE 153	FILE 158	FILE 182	FILE 246				
FILE 219	FILE 211								
FILE 222	FILE 060	FILE 100	FILE 221						
FILE 233	FILE 154								
FILE 235	FILE 170								
FILE 237	FILE 178	FILE 185							
FILE 239	FILE 166								
FILE 242	FILE 176	FILE 180	FILE 188	FILE 195	FILE 202	FILE 208			
FILE 243	FILE 147								
FILE 244	FILE 156	FILE 174							
FILE 247	FILE 245								
FILE 250	FILE 038	FILE 044	FILE 062						
FILE 260	FILE 251	FILE 252	FILE 258						
FILE 264	FINT 025								
FILE 265	FILE 076								
FILE 268	FILE 111	FINT 036							
FILE 280	BSYS 029	FINT 027							
FILE 296	FILE 059	FILE 078							
FILE 301	FILE 283								
FILE 303	FILE 287								
FILE 304	FILE 302								
FILE 305	FILE 285								
FILE 307	FILE 289								
FILE 308	FILE 304	FILE 306							
FILE 311	FILE 299	FILE 300							
FILE 319	BSYS 030	FILE 101	FILE 109	FINT 033	FINT 035				
FILE 325	FILE 105								
FILE 336	FILE 081	FILE 095	FILE 107	FILE 137					

FILT DESCRIPTIVE TEXT

ENTRY POINTS

(0140) THIS IS THE ASSIGNED ADDRESS FOR TRAP ENTRY THE OPERATION BRANCHES ON THE CHAINING, STATUS MODIFIER, AND UNUSUAL CONDITION INDICATORS. THE ROUTINE EITHER READS IN THE NEXT CCW OR BRANCHES TO THE ENDING SEQUENCE.

WRAP THIS ENTRY IS FROM THE BWRP ROUTINE TO SET THE PROGRAM CHECK STATUS BIT.

IPL THIS ENTRY IS MADE DURING THE IPL OPERATION. THE OPERATION LOOPS ON THE FIRST COMMAND UNTIL THE DE TRAP FOR THE RECALIBRATE CLEARS THE LOOP CONDITION. THE OPERATION CONTINUES WITH THE READ DATA SEQUENCE FOR THE INITIAL 24 BYTES.

GETOP THIS IS THE NORMAL ENTRY FOR THE INITIAL SEQUENCE TO OBTAIN THE FIRST CCW. THE CCW IS TESTED FOR VALID FLAGS AND COUNT BEFORE BRANCHING FOR A COMMAND DECODE.

MCHCHK THIS IS A RETURN ENTRY TO THE ROUTINE USED AFTER BRANCHING TO THE MACHINE CHECK TRAP ROUTINE.

SELSTA THIS ENTRY IS USED TO SET THE EQUIPMENT SENSE BIT WHEN AN OP REGISTER FAILURE IS DETECTED.

PTRCHK THIS ENTRY IS FROM THE BWRP ROUTINE TO SET THE PROTECTION CHECK STATUS BIT.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		FILT 001	T			2311 FEATURE	TRAP ROUTINE FOR FILES
		FILT 002	ATABLE	ADDR=0140			
0140	2222	FILT 003				LINK V MMSK1=1	SET FILE PRIORITY-STORE BACK UP
0142	8DAD	FILT 009		011	TRAP N	N=FFI BITS23	
		FILT 010	AEND				
0620	94A4	FILT 011	TRAP	0	071	NOTCHN BR	UNUSUAL CONDITION OR NO CHAINING
0622	94A4	FILT 012	TRAP	1	071	NOTCHN BR	STAT MOD AND UNUS COND OR NO CHN
0624	862A	FILT 013	TRAP	2	016	FRSTBT BR	CMND OR DATA CHAIN-NO UNUS COND
0626	2F7B	FILT 014	TRAP	3		H1=H1+K07	STATUS MOD AND CMND OR DATA CHN
0628	5EE4	FILT 015				H=H+1	ADD 8 TO CCW ADDRESS
062A	1D00	FILT 016	FRSTBT			RST FIA K=80	RESET THE TRAP LATCH
062C	5CF8	FILT 017				RDH P H+2	GET CCW BYTES 0&1
062E	A96D	FILT 018		060	PROCHK N	BR IF P1=NZ	BR IF 2ND BYTE INVALID
0630	0C8B	FILT 019				Z=P0=K08	MASK FOR TIC
0632	FOC2	FILT 020		028	NOTTIC	BR IF LZNZ	BR IF NOT TIC
0634	5EF0	FILT 021				RDH H H+0	GET NEW CCW ADDRESS
0636	0F83	FILT 022				Z=H1*-K08	MASK FOR DOUBLE WORD BOUNDARY
0638	FOEE	FILT 023		067	TICCHK	BR IF LZNZ	BR IF INVALID ADDRESS
063A	5CF8	FILT 024	GETOP			RDH P H+2	GET OP
063C	A96D	FILT 025		060	PROCHK N	BR IF P1=NZ	
063E	0C8B	FILT 026				Z=P0=K08	
0640	C4E1	FILT 027		060	PROCHK 0	BR IF Z=0	
0642	CDC9	FILT 028	NOTTIC	031	CDBR	BR IF FF10=1	BR IF DATA CHAIN

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0644	4FCF	FILT 029				FOP=PO	LOAD THE OP REG
0646	FOE1	FILT 030		060	PRCCHK 0	BR IF LZ=0	BR IF INVALID OP
0648	5AF8	FILT 031	CDBR			RDH T H+2	GET DATA ADDR FROM CCW
064A	5CF8	FILT 032				RDH P H+2	
064C	58F8	FILT 033				RDH I H+2	GET CCW BYTES 6&7
064E	CC83	FILT 034				Z=PO*-K08	MASK FOR VALID FLAG
0650	FOE6	FILT 035		063	PROCHK 3	BR IF LZNZ	BR IF INVALID FLAG-BITS 567 NZ
0652	4886	FILT 036				I=I	TEST FOR VALID CCW COUNT
0654	C4E7	FILT 037		063	PROCHK 3	BR IF Z=0	BR IF INVALID COUNT-COUNT=0000
0656	7E42	FILT 038				STH H DA, 98	SAVE CCW ADDR
0658	CDF5	FILT 039		043	DATCHN	BR IF FFIO=1	BR IF DATA CHAIN
065A	4DCF	FILT 040				FFO=PO	SET CCW FLAGS IN HARDWARE
065C	5C52	FILT 041				RDH P DA, 9A	GET PREVIOUS OP AND FILE MASK
065E	BB98	FILT 042		FILX 007	DECODE N	N=FOPL	BR TO DECODE THE OP
0674	4DCF	FILT 043	DATCHN			FFO=PO	SET FLAGS IN HDWR
0676	2C07	FILT 044				PO=0	SET
0678	2D35	FILT 045				P1=0\$K30	AUX ADDRESS
067A	3D43	FILT 046				P1=P1\$K04	0034
067C	5EC0	FILT 047				RDH H AS, P+0	GET ADDR OF NEXT BYTE-SK SNS DC
067E	8BA3	FILT 048		049	CDRTN N	N=FOP BITS67	
0120	9752	FILT 049	CDRTN 0	FILX 233	LASTST	BR	SENSE OP TEST FOR LAST BYTE
0122	8132	FILT 050	CDRTN 1	058	RDWR	BR	
0124	8132	FILT 051	CDRTN 2	058	RDWR	BR	
0126	FFAE	FILT 052	CDRTN 3	056	TST5	BR IF FOP3=0	CTL OP-BR IF SEEK
0128	DBA0	FILT 053		049	CDRTN 0	BR IF FOP5=0	BR IF SEEK
012A	5C52	FILT 054				RDH P DA, 9A	GET FILE MASK BYTE
012C	B1FE	FILT 055		FILX 088	FMRETN	BR	RETURN TO SET FILE MASK
012E	DBA0	FILT 056	TST5	049	CDRTN 0	BR IF FOP5=0	
0130	CBA0	FILT 057		049	CDRTN 0	BR IF FOP4=0	
0132	5E42	FILT 058	RDWR			RDH H DA, 98	GET CCW ADDR
0134	86F6	FILT 059		FILX 184	CDEND	BR	
0660	6EE4	FILT 060	PROCHK 0			H=H+2	CORRECT THE CCW ADDRESS
0662	6EE4	FILT 061	PROCHK 1			H=H+2	CORRECT THE CCW ADDRESS
0664	6EE4	FILT 062	PROCHK 2			H=H+2	CORRECT THE CCW ADDRESS
0666	7E42	FILT 063	PROCHK 3			STH H DA, 98	SAVE THE CCW ADDRESS
0668	2F25	FILT 064				H1=0\$K20	SET PROGRAM CHECK FOR STATUS
066A	2E07	FILT 065				H0=0	ZERO DEVICF STATUS
066C	8E3A	FILT 066		FILX 412	ZRCSN2	BR	
066E	5E42	FILT 067	TICCHK			RDH H DA, 98	GET CCW ADDR
0670	6EE4	FILT 068				H=H+2	UPDATE CCW ADDR
0672	8660	FILT 069		060	PRCCHK 0	BR	
		FILT 070	ASEQ	AL07=24			
14A4	50FF	FILT 071	NOTCHN			H1=TGRI	GET CPU DETECTED ERRORS
14A6	FCBD	FILT 072		083	NOCPUE	BR IF LZ=0	BR IF NO CPU DETECTED FILE ERROR
14A8	DB2E	FILT 073		076	TSTPRT	BR IF H1 BIT5=0	BR IF NOT MACHINE CHECK
14AA	8220	FILT 074		BMCK 003	TRAPWD	BR	STORE LOG OUT AND RETURN
14AC	2F43	FILT 075	MCHCHK			H1=0\$K04	SET CHAN CTL CHK FOR STATUS
14AE	EB32	FILT 076	TSTPRT	078	TSTWRP	BR IF H1 BIT6=0	BR IF NOT STORAGE PROTECT
14B0	2F15	FILT 077	PRTCHK			H1=0\$K10	SET PROTECTION CHK FOR STATUS
14B2	FB36	FILT 078	TSTWRP	080	RSTBC	BR IF H1 BIT7=0	BR IF NOT STORAGE WRAP
14B4	2F25	FILT 079	WRAP			H1=0\$K20	SET PROGRAM CHK FOR STATUS
14B6	0680	FILT 080	RSTBC			RST BC K=08	RST CPU ERROR LATCHES FOR FILE
14B8	2EC3	FILT 081	DEVSTA			H0=0\$K0C	SET CE+DE FOR STATUS

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
14BA	957A	FILT 082		178	TSTSM	BR	
14BC	2FC7	FILT 083	NOCPU			H1=0	
14BE	C188	FILT 084		081	DEVSTA	BR IF DAS14=0	BR IF NO UNUSUAL CONDITIONS
14C0	E5FE	FILT 085		116	NOTGAT	BR IF DAS12=0	BR IF NOT SEL GATED ATTENTION
14C2	2D45	FILT 086				P1=0\$K40	SET READ GATE BIT
14C4	4EDF	FILT 087				FBO=P1	PUT RD GATE ON BUSS TO FILE
14C6	2E15	FILT 088				H0=0\$K10	SET CTL TAG BIT
14C8	4BEF	FILT 089				TGRO=H0	ISSUE CTL PULSE
14CA	2E43	FILT 090				H0=0\$K04	SET SELECT HEAD BIT
14CC	4BFF	FILT 091				TGRO=H1	RESET TAGS
14CE	1D00	FILT 092				RST FIA K=80	RESET TRAP LATCH
14D0	4EEF	FILT 093				FBO=H0	PUT HD SEL ON BUSS
14D2	FBD6	FILT 094		096	TSTBA	BR IF FOP7=0	BR IF FROM IPL
14D4	B236	FILT 095		FILX 113	NTORST	BR	GET NEXT CCW
14D6	C3FD	FILT 096	TSTBA	115	RTNWD	BR IF BA4=1	BR IF IPL LATCH IS ON
14D8	FD59	FILT 097	IPL	097	IPL	BR IF P1 BIT3=1	WAIT FOR DEVICE END FROM RECAL
14DA	5E42	FILT 098				RDH H DA, 98	GET CCW ADDR
14DC	6EE6	FILT 099				H=H-2	DECREMENT CCW ADDR
14DE	6EE6	FILT 100				H=H-2	TO FLAG BYTE
14E0	5CF8	FILT 101				RDH P H+2	GET FLAGS
14E2	4DCF	FILT 102				FFO=PO	SET FLAGS IN FLAG REG
14E4	2C63	FILT 103				PO=0\$K06	SET READ DATA BITS
14E6	4FCF	FILT 104				FOP=PO	SET READ DATA IN OP REG
14E8	5C52	FILT 105				RDH P DA, 9A	GET PREVIOUS OP IND
14EA	2C07	FILT 106				PO=0	SET PREV OP
14EC	F8EC	FILT 107	WTEIND	107	WTEIND	BR IF FGA7=0	WAIT FOR INDEX
14EE	7C52	FILT 108				STH P DA, 9A	STORE PREVIOUS OP
14F0	0E04	FILT 109				RST FIB K=20	COLD START RESET
14F2	3D00	FILT 110				SET FIA K=80	ISSUE GO PULSE
14F4	2C15	FILT 111				PO=0\$K10	SET CONTROL BIT
14F6	4BCF	FILT 112				TGRO=PO	SET CTL TAG
14F8	6EE4	FILT 113				H=H+2	RESTORE CCW ADDR
14FA	C3FB	FILT 114	WTEBA	114	WTEBA	BR IF BA4=1	IF IPL LATCH ON WAIT HERE
14FC	0222	FILT 115	RTNWD			RTN V MMSK1=0	RETURN FROM TRAP
14FE	CA86	FILT 116	NOTGAT	120	TSTTC	BR IF DS4=0	BR IF NOT TRAP GATE
1500	DF87	FILT 117		120	TSTTC	BR IF FOP1=1	BR IF MULTI-TRACK BIT ON
1502	4BFF	FILT 118				TGRO=H1	RESET TAG REG
1504	A37E	FILT 119		FILX 295	TRPRIN	BR	
1506	5ADF	FILT 120	TSTTC			P1=TC	GET TERMINATING CONDITIONS
1508	C4BC	FILT 121		147	ERRTST	BR IF ZNZ	BR IF ANY ERRORS
150A	5C62	FILT 122				RDH P DA, 9C	GET BUFFERED SEEK DATA
150C	CA9F	FILT 123		132	SWITHD	BR IF DS4=1	BR IF HEAD SWITCH TIME
150E	C5BC	FILT 124		147	ERRTST	BR IF DAS10=0	BR IF NOT COMPARE HA
1510	5BCF	FILT 125				PO=SDI	GET HA FROM READ BUFFER
1512	6CD1	FILT 126				PO=PO^P1	COMPARE HEAD NUMBERS
1514	FC98	FILT 127		129	NOCOMP	BR IF LZNZ	BR IF AT WRONG HEAD
1516	8E0C	FILT 128		FILX 389	SETCTL	BR	
1518	2DC7	FILT 129	NOCOMP			P1=0	ZERO SENSE BYTE 1
151A	2C13	FILT 130	SKCHK			PO=0\$K01	SET SEEK CHK FOR SENSE
151C	956A	FILT 131		170	CHANZ	BR	
151E	5E52	FILT 132	SWITHD			RDH H DA, 9A	GET FILE MASK ALG
1520	EB31	FILT 133		141	FIPRO	BR IF H1 BIT6=1	BR IF FILE PROTECTED
1522	CD32	FILT 134		142	NOSEEK	BR IF P1 BIT0=0	BR IF NO PREV SEEK

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
1524	2D18	FILT 135				P1=P1+K01	ADD 1 TO HEAD NUMBER
1526	7C62	FILT 136				STH P DA,9C	STORE NEW HD
1528	5DF9	FILT 137				H1=P1	
152A	0DAB	FILT 138				Z=P1+K0A	MASK FOR HD 10
152C	F0B5	FILT 139		143	HDSEOC	BR IF LZ=0	BR IF END OF CYL-HD 10
152E	8DD4	FILT 140		FILX 361	GETCYL	BR	
1530	A3D8	FILT 141	FIPRO	FILX 334	PROTEC	BR	
1532	869C	FILT 142	NOSEEK	FILX 139	INVSEQ	BR	
1534	2D25	FILT 143	HDSEOC			P1=0\$K20	SET END OF CYL FOR SENSE
1536	2CC7	FILT 144				P0=0	ZERO SENSE BYTE 0
1538	2FC7	FILT 145				H1=0	ZERO CHAN STATUS
153A	956A	FILT 146		170	CHANZ	BR	
153C	2DC7	FILT 147	ERRTST			P1=0	
153E	2C45	FILT 148				P0=0\$K40	PRESET INTERVENTION REQ SNS
1540	DDEA	FILT 149		170	CHANZ	BR IF DS1=0	BR IF NOT READY
1542	2E85	FILT 150				H0=0\$K80	PRESET UNSAFE SENSE
1544	EED9	FILT 151		161	EQUCHK	BR IF DS2=1	BR IF UNSAFE
1546	FA9B	FILT 152		130	SKCHK	BR IF DS7=1	BR IF SEEK CHECK
1548	5ACF	FILT 153				P0=TC	GET TERMINATING CONDITIONS
154A	C4DC	FILT 154		163	SETSNS	BR IF ZNZ	BR IF ANY ERRORS
154C	7C62	FILT 155				STH P DA,9C	STORE SNS BYTE 2
154E	2EC3	FILT 156				H0=0\$K0C	SET CE+DE FOR STATUS
1550	D8F5	FILT 157		175	SETUEX	BR IF FGA5=1	BR IF UNIT EXCEPTION
1552	C8F7	FILT 158		176	STR9A	BR IF FGA4=1	BR IF ICL
1554	2DC7	FILT 159	SELSTA			P1=0	
1556	2E15	FILT 160				H0=0\$K10	SET U SELECTED STATUS FOR SNS
1558	2C15	FILT 161	EQUCHK			P0=0\$K10	SET EQUIPMENT CHF FOR SNS
155A	956C	FILT 162		171	STR9C	BR	
155C	5CDB	FILT 163	SETSNS			P1=POH	SET SENSE BYTE 1
155E	1CF5	FILT 164				P0=PO*-KFO	STRIP EXTRA BITS FROM SNS BYTE 0
1560	ED64	FILT 165		167	TSTMAM	BR IF P1 BIT2=0	BR IF NOT NO REC FOUND
1562	3D83	FILT 166				P1=P1\$K08	SET NO REC FOUND SENSE
1564	FD68	FILT 167	TSTMAM	169	STRIP	BR IF P1 BIT3=0	BR IF NOT MISSING ADDR MARK
1566	3D23	FILT 168				P1=P1\$K02	SET MISS AM FOR SENSE
1568	1D35	FILT 169	STRIP			P1=P1*-K30	STRIP EXTRA BITS FROM SNS BYTE 1
156A	2EG7	FILT 170	CHANZ			H0=0	ZERO SNS BYTE 2&CHAN STATUS
156C	7E62	FILT 171	STR9C			STH H DA,9C	STORE SNS BYTE 2
156E	2EE3	FILT 172				H0=0\$K0E	SET CE+DE+UC FOR STATUS
1570	3C25	FILT 173				P0=PO\$K20	SET CONTINGENT CONNECTION BIT
1572	D8F6	FILT 174		176	STR9A	BR IF FGA5=0	BR IF NO UNIT EXCEPTION
1574	3E13	FILT 175	SETUEX			H0=H0\$K01	SET UNIT EXCEPTION IN STATUS
1576	7C52	FILT 176	STR9A			STH P DA,9A	STORE SNS BYTES 0&1
1578	2F07	FILT 177				H1=0	ZERO CHAN STATUS
157A	FDFE	FILT 178	TSTSM	180	TSTPCI	BR IF FF13=0	BR IF NO STATUS MOD
157C	3E45	FILT 179				H0=H0\$K40	SET STAT MOD STATUS
157E	C982	FILT 180	TSTPCI	182	RSTTRP	BR IF FF14=0	BR IF NOT PCI
1580	3F85	FILT 181				H1=H1\$K80	SET PCI IN STATUS
1582	1D04	FILT 182	RSTTRP			RST FIA K=A0	RESET PCI AND TRAP LATCHES
1584	C88D	FILT 183		187	SETICL	BR IF FGA4=1	BR IF INCORRECT LENGTH
1586	F98F	FILT 184		188	ZERT	BR IF FF17=1	BR IF SLI
1588	4886	FILT 185				I=1	TEST FOR CCW COUNT 0
158A	C48F	FILT 186		188	ZERT	BR IF Z=0	BR IF LENGTH OK
158C	3F45	FILT 187	SETICL			H1=H1\$K40	SET ICL FOR STATUS

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
158E	2AC7	FILT 188	ZERT			T0=0	
1590	CF83	FILT 189				Z=H1*-K08	TST FOR CHAN ERROR STATUS
1592	C496	FILT 190		192	SETAUX	BR IF ZN3	BR IF HAD ERROR
1594	EA20	FILT 191		197	PASS	BR IF H0 BIT6=0	BR IF NO UNIT CHECK
1596	2B43	FILT 192	SETAUX			T1=0\$K04	SET AUX ADDR 0014 T/
1598	3B15	FILT 193				T1=T1\$K10	ADDR CE AID BUFFER
159A	5C42	FILT 194				RDH P DA,98	GET FAILING CCW ADDR+8
159C	7EA8	FILT 195				STH H AS,T+2	STORE ERROR STATUS IN AUX 0014
159E	7CA0	FILT 196				STH P AS,T+0	STORE ERROR CCW ADDR IN AUX 0016
15A0	8E40	FILT 197	PASS	FILX 415	CHNEND	BR	
		FILT 198	AEND				

 * CROSS REFERENCE FOR CSECT FILT *

FILT 011	FILT 0C9						
FILT 016	FILT 013	FILX 087	FILX 116				
FILT 024	FILT 231						
FILT 028	FILT 020						
FILT 031	FILT 028						
FILT 043	FILT 039						
FILT 049	FILT 048	FILT 053	FILT 056	FILT 057			
FILT 056	FILT 052						
FILT 058	FILT 050	FILT 051					
FILT 060	FILT 018	FILT 025	FILT 027	FILT 030	FILT 069		
FILT 063	FILT 035	FILT 037					
FILT 067	FILT 023						
FILT 071	FILT 011	FILT 012					
FILT 075	BMCK 038						
FILT 076	FILT 073						
FILT 077	BWRP 022						
FILT 078	FILT 076						
FILT 079	BWRP 013						
FILT 080	FILT 078						
FILT 081	FILT 084						
FILT 083	FILT 072						
FILT 096	FILT 054						
FILT 097	FILT 057	FILX 329					
FILT 107	FILT 1C7						
FILT 114	FILT 114						
FILT 115	FILT 056						
FILT 116	FILT 085						
FILT 120	FILT 116	FILT 117					
FILT 129	FILT 127						
FILT 130	FILT 152						
FILT 132	FILT 123						
FILT 141	FILT 133						
FILT 142	FILT 134						
FILT 143	FILT 139						
FILT 147	FILT 121	FILT 124					
FILT 159	FILX 0C7						
FILT 161	FILT 151						
FILT 163	FILT 154						
FILT 167	FILT 165						

* CROSS REFERENCE FOR CSECT FILT *

FILT 169	FILT 167		
FILT 170	FILT 131	FILT 146	FILT 149
FILT 171	FILT 162		
FILT 175	FILT 157		
FILT 176	FILT 158	FILT 174	
FILT 178	FILT 082		
FILT 180	FILT 178		
FILT 182	FILT 180		
FILT 187	FILT 183		
FILT 188	FILT 184	FILT 186	
FILT 192	FILT 190		
FILT 197	FILT 191		

FILX DESCRIPTIVE TEXT

ENTRY POINTS

<p>DECODE N THIS ENTRY IS FROM THE FILT ROUTINE TO DECODE THE COMMAND. A SIXTEEN WAY BRANCH MAKES THE INITIAL COMMAND SELECTION ON THE FOUR LOW ORDER BITS. FINAL SELECTION AND VERIFICATION OF EACH COMMAND IS MADE BEFORE PROCEEDING TO THE EXECUTION SEQUENCE.</p> <p>NTORST THIS ENTRY PROVIDES A RESET OF THE NTO LATCH TO ALLOW ORIENTING THE FILE FOR THE NEXT COMMAND. IT IS USED FOLLOWING THE DE TESTING FOR A CONTROL COMMAND.</p> <p>INVSEQ THIS ENTRY SETS THE INVALID SEQUENCE AND COMMAND REJECT SENSE BITS. IT IS USED FROM VARIOUS POINTS THAT DETECT COMMAND SEQUENCE ERRORS.</p> <p>GET8E THIS IS AN ENTRY USED TO END THE MICRO-ROUTINE FOR THE COMMAND. IF THE OPERATION ORIGINALLY ENTERED THE ROUTINE DURING THE INITIAL SEQUENCE, THE ACTIVE BIT IS SET AND THE OPERATION RETURNED TO ICYCLES. WHEN THE ROUTINE IS ENTERED THROUGH THE TRAP, THE OPERATION ENDS WITH RETURN FROM TRAP.</p> <p>TRPRTN THIS ENTRY IS USED TO RESTART A RECALIBRATE OR SEEK OPERATION THAT WAS STOPPED BECAUSE THE FILE WAS STILL ERASING. THE EXIT WAS MADE FROM THE ROUTINE WITHCUT RESETTING THE COMMAND DATA.</p>	<p>PROTEC THIS ENTRY IS USED TO SET THE CONTINGENT CONNECTION INDICATOR AND THE FILE PROTECT SENSE BIT BEFORE SETTING THE SENSE BYTES TO END THE OPERATION.</p> <p>GETCYL THIS ENTRY IS USED TO ENTER THE SEEK ROUTINE FOR A HEAD ADVANCE DURING A MULTIPLE TRACK READ OR SEARCH OPERATION.</p> <p>SETCYL THIS ENTRY IS USED TO REINITIATE THE READ OR SEARCH OPERATION AFTER TESTING FOR A SUCCESSFUL HEAD COMPARE FOLLOWING A MULTIPLE TRACK HEAD ADVANCE.</p> <p>ZROSN2 THIS IS AN ENTRY INTO THE CHAIN END SEQUENCE ROUTINE THAT FIRST RESETS SENSE BYTE 2 BEFORE ENTERING THE NORMAL ENDING SEQUENCE.</p> <p>CHNEND THIS IS THE NORMAL CHAIN END SEQUENCE ENTRY USED TO SET THE INTERRUPT BUFFER AND THE INTERRUPT LATCH. THE RETURN FROM TRAP OCCURS AFTER RESETTING THE FILE CONTROL OP AND FLAG REGISTERS.</p>
--	--

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		FILX 001	T			2311 FEATURE COMMAND VERIFICATION AND EXECUTION	
		FILX 002	*			*****	
		FILX 003	*			*	**
		FILX 004	*			FILE OP DECODE - COMMON	**
		FILX 005	*			*	**
		FILX 006	*			*****	
0580	9554	FILX 007	DECODE 0	FILT 159	SELSTA	BR	OP REG FAILURE SET EQUIP CHK
0582	8FFD	FILX 008	DECODE 1	173	ERASE N	N=FOP BITS23	ERASE OR SEARCH ID
0584	EFB8	FILX 009	DECODE 2	049	IPLRCT	BR IF FOP2=0	READ IPL OR READ COUNT
0586	EFD6	FILX 010	DECODE 3	069	CTLOP	BR IF FOP2=0	NO OP OR RECALIBRATE

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS	
0588	9774	FILX 011	DECODE 4	195	SENSE	BR	SENSE OP	
058A	8F9D	FILX 012	DECODE 5	125	WRDARO N	N=FOP BITS23	WRITE DATA OR RO	
058C	EFCC	FILX 013	DECODE 6	059	RDDARO	BR IF FOP2=0	READ DATA OR RO	
058E	EFD6	FILX 014	DECODE 7	069	CTLOP	BR IF FOP2=0	SEEK OR RESTORE	
0590	86AC	FILX 015	DECODE 8	147	INVCMN	BR	INVALID CMND	
0592	8FBD	FILX 016	DECODE 9	141	WRHA N	N=FOP BITS23	WRITE HA OR SEARCH KEY	
0594	EFCE	FILX 017	DECODE A	057	RDHA	BR IF FOP2=0	READ HA	
0596	EFD6	FILX 018	DECODE B	069	CTLOP	BR IF FOP2=0	SEEK CYL OR SEEK HD	
0598	86AC	FILX 019	DECODE C	147	INVCMN	BR	INVALID CMND	
059A	8FDD	FILX 020	DECODE D	157	WRKD N	N=FOP BITS23	WRITE KEY & DATA OR SCAN	
059C	EFCE	FILX 021	DECODE E	060	RDCKD	BR IF FOP2=0	READ COUNT KEY DATA	
059E	EF91	FILX 022	DECODE F	015	DECODE 8	BR IF FOP2=1	SET FILE MASK OR SPACE COUNT	
05A0	DF91	FILX 023		015	DECODE 8	BR IF FOP1=1	BR IF INVALID COMMAND	
05A2	5E32	FILX 024				RDH H DA,8E	GET STATS	
05A4	FFF2	FILX 025		029	SPACE	BR IF FOP3=0	BR IF SPACE COUNT OP	
05A6	3E00	FILX 026				SET FIB K=80	SET NON TIME ORIENTED	
05A8	CD2C	FILX 027		038	RMVSIO	BR IF P1 BIT0=0	BR IF SET FILE MASK ALLOWED	
05AA	869C	FILX 028	SEQ	139	INVSEQ	BR		
05F2	DE2D	FILX 029	SPACE	038	RMVSIO	BR IF H0 BIT1=1	BR IF FIRST OP IN CHAIN	
05F4	F5AB	FILX 030		028	SEQ	BR IF DAS13=1	BR IF ERASE GATE ON	
05F6	EC31	FILX 031		040	PIND	BR IF P0 BIT2=1	BR IF CHND FRM SNS,CTL	
05F8	2C33	FILX 032				PO=0\$K03		
05FA	4FCF	FILX 033				FOP=P0	STRIP KEY&DATA BITS FROM OP REG	
05FC	5DDD	FILX 034				P1=P1L	SAVE SEEK MASK	
05FE	3DB5	FILX 035				P1=P1\$KBO	SET MASK TO INH ALL WRITES	
0600	5E42	FILX 036				RDH H DA,98	GET CCW ADDR	
0602	86EA	FILX 037		178	RSPOGO	BR	SET PREV OP AND GO PULSE	
05AC	1E45	FILX 038	RMVSIO			H0=H0*-K40	STRIP CAME FROM SID STAT	
05AE	7E32	FILX 039				STH H DA,8E	STORE STATS	
05B0	2CC7	FILX 040	PIND			PO=0	SET PREVIOUS OP	
05B2	FFE1	FILX 041		079	SETFM	BR IF FOP3=1	BR IF SET FILE MASK OP	
05B4	F8B4	FILX 042	SCINDX	042	SCINDX	BR IF FGA7=0	WAIT FOR INDEX	
05B6	B228	FILX 043		106	STRFM	BR	GO STORE PREV OP AND END	
		FILX 044	*	*****				
		FILX 045	*	*			**	
		FILX 046	*	*	IPL OR READ COUNT		**	
		FILX 047	*	*			**	
		FILX 048	*	*****				
05B8	FFCB	FILX 049	IPLRCT	058	SETPO	BR IF FOP3=1	BR IF READ COUNT OP	
05BA	DF91	FILX 050		015	DECODE 8	BR IF FOP1=1	BR IF INVALID CMND	
05BC	CD11	FILX 051		015	DECODE 8	BR IF P1 BIT0=1	BR IF PREVIOUS SET FILE MASK IPL	
05BE	3E00	FILX 052				SET FIB K=80	SET NON TIME ORIENTED	
05C0	F5C1	FILX 053	IPLRD	053	IPLRD	BR IF DAS13=1	BR IF ERASE GATE ON	
05C2	2C65	FILX 054				PO=0\$K60	TURN ON CC AND SLI	
05C4	4DCF	FILX 055				FF0=P0	SET CNND CHN IN FLAG REG	
05C6	A39A	FILX 056		315	RECAL	BR	GO EXECUTE RECALIBRATE	
05C8	FF90	FILX 057	RDHA	015	DECODE 8	BR IF FOP3=0	BR IF INVALID CMND	
05CA	86EA	FILX 058	SETPO	178	RSPOGO	BR		
05CC	E851	FILX 059	RDDARO	061	SMTST	BR IF P0 BIT6=1	BR IF PREVIOUS SEARCH	
05CE	F84A	FILX 060	RCKD	058	SETPO	BR IF P0 BIT7=0	BR IF NO PREV SEARCH	
05D0	FFCB	FILX 061	SMTST	058	SETPO	BR IF FOP3=1	BR IF READ RO OR READ CKD	
05D2	D1CA	FILX 062		058	SETPO	BR IF DAS15=0	BR IF NO STATUS MODIFIER	
05D4	8694	FILX 063		135	SETO4	BR		

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		FILX 064	*			*****	
		FILX 065	*			*	**
		FILX 066	*			CONTROL OPS (NTO RESET)	**
		FILX 067	*			*	**
		FILX 068	*			*****	
05D6	DF91	FILX 069	CTLOP	015	DECODE 8	BR IF FOP1=1	BR IF INVALID CMND
05D8	3E00	FILX 070				SET FIB K=80	SET NON TIME ORIENTED LATCH
05DA	2C25	FILX 071				PO=0\$K20	SET PREV OP IND
05DC	7C52	FILX 072				STH P DA,9A	STORE PREV OP
05DE	A384	FILX 073		298	CTLS	BR	
		FILX 074	*			*****	
		FILX 075	*			*	**
		FILX 076	*			SET FILE MASK	**
		FILX 077	*			*	**
		FILX 078	*			*****	
05E0	50B8	FILX 079	SETFM			RDB P1 T+1	GET FILE MASK BYTE
05E2	CDE9	FILX 080		083	DCIDEC	BR IF FFI0=1	BR IF DATA CHAIN
05E4	5886	FILX 081				I=I-1	DECREMENT CCW COUNT
05E6	B1FE	FILX 082	IGNRDC	088	FMRETN	BR	CONTINUE WITH SET FILE MASK
05E8	5886	FILX 083	DCIDEC			I=I-1	DECREMENT CCW COUNT
05EA	C4E6	FILX 084		082	IGNRDC	BR IF ZNZ	BR IF COUNT NOT ZERO
05EC	7C52	FILX 085				STH P DA,9A	SAVE FILE MASK BYTE
05EE	5E42	FILX 086	CCWADR			RDH H DA,98	RESTORE CCW ADDRESS IN H
05F0	862A	FILX 087		FILT 016	FRSTBT	BR	GET NEXT CCW FOR DATA CHAIN
31FE	0D83	FILX 088	FMRETN			Z=P1*-K08	TEST FOR VALID MASK LO
3200	FC84	FILX 089		091	INVFM	BR IF LZNZ	BR IF MASK INVALID
3202	ED0C	FILX 090		092	NONSNS	BR IF P1 BIT2=0	BR IF VALID MASK HI
3204	97A6	FILX 091	INVFM	273	CHNSET	BR	
320C	3D13	FILX 092	NONSNS			P1=P1\$K01	SET NONSENSE STORED BIT
320E	2C25	FILX 093				PO=0\$K20	SET PREVIOUS OP
3210	C916	FILX 094		097	TSTB3	BR IF P1 BIT4=0	BR IF INH ALL\$ALLOW ALL SEEKS
3212	FD1C	FILX 095		100	TSTB0	BR IF P1 BIT3=0	MASK ALLOWS ALL SEEKS 1ST WRITES
3214	3DF3	FILX 096				P1=P1\$K0F	SET ALG TO INH ALL SEEKS
3216	FD1C	FILX 097	TSTB3	100	TSTB0	BR IF P1 BIT3=0	BR IF ALLOW ALL SEEKS
3218	3DD3	FILX 098				P1=P1\$K0D	SET ALG TO INH SEEK&SEEK CYL
321A	1D15	FILX 099				P1=P1*-K10	REMOVE BIT 3 OF ALGORITHM
321C	CD25	FILX 100	TSTB0	104	TSTB1	BR IF P1 BIT0=1	BR IF ALLOW ALL\$INH RO-HA-CKD
321E	3D85	FILX 101				P1=P1\$K80	SET ALG TO INH SET FILE MASK
3220	DD28	FILX 102		106	STRFM	BR IF P1 BIT1=0	BR IF ALGORITHM COMPLETE
3222	1DFD	FILX 103				P1=P1\$KFO	SETS ALG TO INHIBIT ALL WRITES
3224	DD29	FILX 104	TSTB1	106	STRFM	BR IF P1 BIT1=1	BR IF ALG IS COMPLETE
3226	3DA5	FILX 105				P1=P1\$KA0	SET ALG TO INH WR HA-RO-CKD
3228	7C52	FILX 106	STRFM			STH P DA,9A	STORE FILE MASK AND PREV OP ALGS
322A	2F45	FILX 107	COMEND			H1=0\$K40	PRESET INCORRECT LENGTH BIT
322C	CD89	FILX 108		118	STASET	BR IF FFI0=1	BR IF DATA CHAIN
322E	F9B5	FILX 109		112	CORLEN	BR IF FFI7=1	BR IF SLI ON
3230	4886	FILX 110				I=I	TEST FOR CCW COUNT 0
3232	C488	FILX 111		118	STASET	BR IF ZNZ	BR IF INCORRECT LEGNTH
3234	DD86	FILX 112	CORLEN	117	CEADE	BR IF FFI1=0	BR IF NOT COMMAND CHAIN
3236	2E08	FILX 113	NTORST			SET FIB K=40	SER FLAG BITS TO ZERO
3238	GE04	FILX 114	CLDRST			RST FIB K=20	COLD START RESET
323A	5E42	FILX 115				RDH H DA,98	GET CCW ADDR
323C	862A	FILX 116		FILT 016	FRSTBT	BR	GET NEW CCW

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
3206	2F07	FILX 117	CEADE			H1=0	ZERO CHAN STATUS
3208	2EC3	FILX 118	STASET			H0=0\$K0C	SET CHAN END AND DEVICE END
320A	8E3A	FILX 119		412	ZROSN2	BR	
		FILX 120	*				*****
		FILX 121	*				**
		FILX 122	*			WRITE DATA OR RO (05 OR 15)	**
		FILX 123	*				**
		FILX 124	*				*****
0680	DF88	FILX 125	WRDARO 0	129	WRDATA	BR IF FOP1=0	BR IF WRITE DATA OP
0682	DF8E	FILX 126	WRDARO 1	132	WRR0OP	BR IF FOP1=0	BR IF WRITE RO OP
0684	86AC	FILX 127	WRDARO 2	147	INVCMN	BR	INVALID COMMAND DECODED
0686	86AC	FILX 128	WRDARO 3	147	INVCMN	BR	INVALID COMMAND DECODED
0688	2D10	FILX 129	WRDATA			SET FIA K=01	SET READ CLOCK CONTROL
068A	E833	FILX 130		150	DASIS	BR IF P0 BIT6=1	BR IF PREVIOUS SEARCH ID OR KEY
068C	869C	FILX 131		139	INVSEQ	BR	INVALID SEQUENCE
068E	C81B	FILX 132	WRR0OP	138	TDASI	BR IF P0 BIT4=1	BR IF PREVIOUS SEARCH HA
0690	FC1C	FILX 133		139	INVSEQ	BR IF P0 BIT3=0	BR IF NOT PREV WRITE HA
0692	DD36	FILX 134	PITEST	152	FIPROT	BR IF P1 BIT1=0	BR IF FILE PROTECTED
0694	3D00	FILX 135	SET04			SET FIA K=80	ISSUE GO PULSE
0696	2C43	FILX 136				P0=0\$K04	SET PREVIOUS OP
0698	86EE	FILX 137		180	STPO	BR	
069A	D193	FILX 138	TDASI	134	PITEST	BR IF DASIS=1	BR IF STATUS MOD&NOT SHORT SCH
069C	2F15	FILX 139	INVSEQ			H1=0\$K10	SET INVALID SEQ BIT FOR SENSE
069E	86B8	FILX 140		153	COMREJ	BR	
06A0	86AC	FILX 141	WRHA 0	147	INVCMN	BR	
06A2	86D8	FILX 142	WRHA 1	169	WRHAOP	BR	WRITE HOME ADDRESS
06A4	2C23	FILX 143	WRHA 2			P0=0\$K02	PRESET SEARCH KEY ALGORITHM
06A6	FFE8	FILX 144	WRHA 3	177	TSTOP	BR IF FOP3=0	BR IF SEARCH KEY OP
06A8	2C83	FILX 145				P0=0\$K08	PRESET SEARCH HA ALGORITHM
06AA	CFEC	FILX 146		179	DATAOP	BR IF FOP0=0	BR IF SCH HA EQUAL
06AC	2F07	FILX 147	INVCMN			H1=0	ZERO SENSE BYTE 1
06AE	86B8	FILX 148		153	COMREJ	BR	
06B0	F81C	FILX 149	WRKDOP	139	INVSEQ	BR IF P0 BIT7=0	BR IF NO PREV SEARCH
06B2	D19C	FILX 150	DASIS	139	INVSEQ	BR IF DASIS=0	BR IF NO STATUS MOD
06B4	FD6A	FILX 151		178	RSPOGO	BR IF P1 BIT3=0	BR IF NO FILE PROTECT
06B6	2F43	FILX 152	FIPROT			H1=0\$K04	SET FILE PROTECT FOR SENSE
06B8	2EA5	FILX 153	COMREJ			H0=0\$KA0	SET CMND REJ&CONTINGENT CONN SNS
06BA	7E52	FILX 154	SETSEN			STH H DA, 9A	STORE SENSE BYTES 0&1
06BC	2E23	FILX 155				H0=0\$K02	SET UNIT CHECK FOR STATUS
06BE	8E38	FILX 156		411	ZROSTA	BR	
06C0	DFB0	FILX 157	WRKD 0	149	WRKDOP	BR IF FOP1=0	BR IF WRITE KEY DATA OP
06C2	DFC8	FILX 158	WRKD 1	161	WRCKD	BR IF FOP1=0	BR IF WRITE COUNT KEY DATA OP
06C4	EFEB	FILX 159	WRKD 2	178	RSPOGO	BR IF FOP2=1	BR IF SCAN OP
06C6	86AC	FILX 160	WRKD 3	147	INVCMN	BR	
06C8	E89D	FILX 161	WRCKD	139	INVSEQ	BR IF FGA6=1	BR IF ON DEFECTIVE TRACK
06CA	E853	FILX 162		166	SETCKD	BR IF P0 BIT6=1	BR IF PREV SCH ID OR KEY
06CC	D81C	FILX 163		139	INVSEQ	BR IF P0 BIT5=0	BR IF NOT PREV WRITE CKD
06CE	ED14	FILX 164	P2TEST	135	SET04	BR IF P1 BIT2=0	BR IF NOT FILE PROTECT
06D0	86B6	FILX 165		152	FIPROT	BR	
06D2	2D1C	FILX 166	SETCKD			SET FIA K=01	SET READ CLOCK CONTROL
06D4	D1CF	FILX 167		164	P2TEST	BR IF DASIS=1	BR IF STATUS MOD&NOT SHORT SCH
06D6	869C	FILX 168		139	INVSEQ	BR	
06D8	DFAD	FILX 169	WRHAOP	147	INVCMN	BR IF FOP1=1	BR IF INVALID COMMAND

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
06DA	2C15	FILX 170				PO=0\$K10	SET WRITE HA ALGORITHM
06DC	DD6D	FILX 171		179	DATAOP	BR IF P1 BIT1=1	BR IF NOT FILE PROTECT
06DE	86B6	FILX 172		152	FIPROT	BR	
06E0	86AC	FILX 173	ERASE 0	147	INVCMN	BR	INVALID COMMAND***WRITE SPECIAL
06E2	DFC8	FILX 174	ERASE 1	161	WRCKD	BR IF FOP1=0	BR IF WRITE COUNT KEY DATA OP
06E4	86AC	FILX 175	ERASE 2	147	INVCMN	BR	INVALID COMMAND DECODED
06E6	2C33	FILX 176	ERASE 3			PO=0\$K03	SET SEARCH ID ALGORITHM
06E8	CFEC	FILX 177	TSTOP	179	DATAOP	BR IF FOP0=0	BR IF SEARCH EQUAL
06EA	2CC7	FILX 178	RSPOGO			PO=0	SET PREVIOUS OP ALG
06EC	3D00	FILX 179	DATAOP			SET FIA K=80	ISSUE GO PULSE
06EE	7C52	FILX 180	STPO			STH P DA,9A	STORE PREVIOUS OP ALG
06F0	2C15	FILX 181				PO=0\$K10	SET CONTROL BIT
06F2	4BCF	FILX 182				TGRD=PO	RAISE CONTROL TAG
06F4	1D00	FILX 183	GET8E			RST FIA K=80	RESET TRAP LATCH
06F6	5C32	FILX 184	CDEND			RDH P DA,8E	GET STATS
06F8	EC7D	FILX 185		187	RTNIC	BR IF P0 BIT2=1	BR IF RETURN TO I CYCLES
06FA	0222	FILX 186				RTN V MMSK1=0	RETURN FROM TRAP
06FC	2C85	FILX 187	RTNIC			PO=0\$K80	SET ACTIVE STAT
06FE	7C32	FILX 188				STH P DA,8E	STORE STATS
0700	B362	FILX 189		CCOM 154	CCOB	BR	SET CCO
		FILX 190	*				*****
		FILX 191	*				**
		FILX 192	*			SENSE & SEEK-SENSE COMMON	**
		FILX 193	*				**
		FILX 194	*				*****
1774	3E00	FILX 195	SENSE			SET FIB K=80	SET NON TIME ORIENTED
1776	5FFF	FILX 196				H1=FOP	GET OP
1778	E0DD	FILX 197		199	SNSEX	BR IF HZ=0	BR IF VALID OP
177A	86AC	FILX 198		147	INVCMN	BR	
175C	1C25	FILX 199	SNSEX			PO=PO*-K20	STRIP CONTINGENT CONNECTION BIT
175E	7C52	FILX 200				STH P DA,9A	SAVE ALTERED SENSE
1760	5E32	FILX 201	SKSNS			RDH H DA,8E	GET STATS
1762	1E45	FILX 202				HO=HO*-K40	REMOVE CAME FROM SID STAT
1764	7E32	FILX 203				STH H DA,8E	SAVE STATS
1766	2E07	FILX 204				HO=0	SET
1768	2F25	FILX 205				H1=0\$K20	AUX ADDRESS
176A	3F43	FILX 206				H1=H1\$K04	0024
176C	FBA6	FILX 207		211	SNSRD	BR IF FOP7=0	BR IF SENSE OP
176E	5DB8	FILX 208	SKRD			RDB P1 T+1	GET A BYTE OF SEEK DATA
1770	7DE8	FILX 209				STB P1 AS,H+1	SAVE BYTE IN AUX
1772	974E	FILX 210		231	SKIPON	BR	
1726	F92E	FILX 211	SNSRD	215	STRO1	BR IF P1 BIT7=0	BR IF SENSE IS STORED
1728	2C07	FILX 212				PO=0	ZERO
172A	2DC7	FILX 213				P1=0	SENSE
172C	7C62	FILX 214				STH P DA,9C	BYTES
172E	7CE8	FILX 215	STRO1			STH P AS,H+2	STORE SNS BYTES 0&1 IN AUX
1730	5C62	FILX 216				RDH P DA,9C	GET SNS BYTE 2
1732	5EDF	FILX 217				P1=DS	SET SENSE BYTE 3
1734	1DA3	FILX 218				P1=P1*-K0A	STRIP BITS 4,6
1736	DD3A	FILX 219		221	DUMPSN	BR IF P1 BIT1=0	BR IF NOT ON LINE
1738	3D83	FILX 220				P1=P1\$K08	OR IN ON LINE BIT
173A	7CE0	FILX 221	DUMPSN			STH P AS,H+0	
173C	2CC7	FILX 222				PO=0	ZERO SENSE

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
173E	2DC7	FILX 223				P1=0	BYTES 4&5
1740	3F15	FILX 224				H1=H1\$K10	SET AUX ADDR 0034
1742	7CEA	FILX 225				STH P AS,H-2	STORE BYTES 4&5 IN AUX
1744	1F1D	FILX 226	ADRINK			H1=H1\$K10	SET AUX ADDR FOR NEXT BYTE
1746	FBF0	FILX 227	TSTFOP	208	SKRD	BR IF FOP7=1	BR IF SEEK OP
1748	5DE8	FILX 228				RDB P1 AS,H+1	GET AS SENSE BYTE
174A	05CF	FILX 229		231	SKIPON	BR IF DAS11=1	BR IF SKIP IS ON
174C	7DB8	FILX 230				STB P1 T+1	STORE A SENSE BYTE
174E	5886	FILX 231	SKIPON			I=I-1	DECREMENT CCW COUNT
1750	C4FD	FILX 232		239	TSTCTR	BR IF Z=0	BR IF COUNT AT ZERO
1752	CB46	FILX 233	LASTST	227	TSTFOP	BR IF H1 BIT4=0	BR IF NOT LAST BYTE
1754	1FEB	FILX 234				H1=H1\$K0E	RESTORE LO AUX ADDR
1756	FF44	FILX 235		226	ADRINK	BR IF H1 BIT3=0	BR IF NOT LAST BYTE
1758	FBA5	FILX 236		238	SKX	BR IF FOP7=1	BR IF SEEK
175A	B22A	FILX 237		107	COMEND	BR	
1724	978E	FILX 238	SKX	261	SEEKOP	BR	
177C	CD9B	FILX 239	TSTCTR	248	SETADR	BR IF FF10=1	BR IF DATA CHAIN
177E	CB06	FILX 240		244	SHORT	BR IF H1 BIT4=0	BR IF NOT LAST BYTE
1780	FF06	FILX 241		244	SHORT	BR IF H1 BIT3=0	BR IF NOT LAST BYTE
1782	FB8F	FILX 242		261	SEEKOP	BR IF FOP7=1	BR IF SEEK
1784	B22A	FILX 243	SLIISN	107	COMEND	BR	
1786	2F45	FILX 244	SHORT			H1=0\$K40	PRESETN INCORRECT LENGTH
1788	FBC9	FILX 245		253	SKERR	BR IF FOP7=1	BR IF SEEK OP
178A	F985	FILX 246		243	SLIISN	BR IF FF17=1	BR IF SLI
178C	B208	FILX 247		118	STASET	BR	
171A	2C07	FILX 248	SETADR			P0=0	SET AUX
171C	2D35	FILX 249				P1=0\$K30	ADDRESS
171E	3D43	FILX 250				P1=P1\$K04	0034
1720	7EC0	FILX 251				STH H AS,P+0	STORE NEXT BYTE ADDR FOR DATA CH
1722	85EE	FILX 252		086	CCWADR	BR	
1708	2C13	FILX 253	SKERR			P0=0\$K01	SET SEEK CHECK FOR SNS
170A	F9CE	FILX 254		256	CMREJ	BR IF FF17=0	BR IF NOT SLI
170C	2FC7	FILX 255	CHNZER			H1=0	ZERO CHAN STATUS
170E	3CA5	FILX 256	CMREJ			P0=P0\$KAO	SET CMND REJECT&CONTIN CONN BITS
1700	2DC7	FILX 257				P1=0	ZERO SNS BYTE 1
1702	7C52	FILX 258				STH P DA,9A	STORE SNS
1704	2EE3	FILX 259				H0=0\$K0E	SET CE&DE&UC
1706	8E3A	FILX 260		412	ZRCSN2	BR	
178E	2A07	FILX 261	SEEKOP			T0=0	SET AUX
1790	2B25	FILX 262				T1=0\$K20	ADDRESS
1792	3B43	FILX 263				T1=T1\$K04	0024
1794	5CA8	FILX 264				RDH P AS,T+2	GET BYTES 0&1 OF SEEK DATA
1796	4CC6	FILX 265				P=P	
1798	C4A4	FILX 266		272	SKCHK	BR IF ZNZ	BR IF BYTES ARE NON ZERO
179A	5CA0	FILX 267				RDH P AS,T+0	GET BYTES 2&3 OF SEEK DATA
179C	3B15	FILX 268				T1=T1\$K10	SET AUX ADDR 36
179E	5EAA	FILX 269				RDH H AS,T-2	GET BYTES 4&5 OF SEEK DATA
17A0	6EC5	FILX 270				H0=H0\$P0	MASK BYTES 2&4 FOR ZERO
17A2	C4AF	FILX 271		280	SAVCYL	BR IF Z=0	BR IF BYTES 2&4 ARE VALID
17A4	2C13	FILX 272	SKCHK			P0=0\$K01	SET SEEK CHECK FOR SENSE
17A6	2F45	FILX 273	CHNSET			H1=0\$K40	SET WLR
17A8	CDC7	FILX 274		279	ILDN	BR IF FF10=1	BR IF DATA CHAIN
17AA	F9C2	FILX 275		277	TSTI	BR IF FF17=0	BR IF NOT SLI

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
17AC	97CC	FILX 276	NOIL	255	CHNZER	BR	
17C2	48E6	FILX 277	TSTI			I=I	MASK FOR CCW COUNT ZERO
17C4	C4AD	FILX 278		276	NOIL	BR IF Z=0	BR IF COUNT ZERO
17C6	97CE	FILX 279	ILON	256	CMREJ	BR	
17AE	5DE9	FILX 280	SAVCYL			HO=P1	SAVE CYL NO
17B0	3B13	FILX 281				T1=T1\$K01	SET CONSTANT 35
17B2	6BE3	FILX 282				T1=T1+HO	MASK FOR VALID CYL NO
17B4	F4BA	FILX 283		286	TSTHD	BR IF AC=0	BR IF VALID CYL
17B6	0EFF	FILX 284				Z=HO\$KFF	MASK FOR CYL 255
17B8	C4A4	FILX 285		272	SKCHK	BR IF ZNZ	BR IF INVALID CYL
17BA	2CA3	FILX 286	TSTHD			PO=0\$K0A	SET CONSTANT
17BC	7CF3	FILX 287				PO=PO-HI	MASK FOR VALID HD
17BE	F4A4	FILX 288		272	SKCHK	BR IF AC=0	BR IF INVALID HD
17C0	8DCE	FILX 289		358	FWDSET	BR	
		FILX 290	*				*****
		FILX 291	*				**
		FILX 292	*				**
		FILX 293	*				**
		FILX 294	*			RECALIBRATE OR SEEK, AFTER ERASE IS FINISHED	**
		FILX 295	*				*****
237E	1D00	FILX 295	TRPRTN			RST FIA K=80	RESET TRAP LATCH
2380	E964	FILX 296		304	CETIME	BR IF P1 BIT6=0	BR IF CHANNEL END TIME FOE RECAL
2382	5C52	FILX 297				RDH P DA,9A	GET FILE MASK ALGORITHM
2384	F590	FILX 298	CTLS	310	EXCTLS	BR IF DASI3=0	BR IF NOT ERASE GATE
2386	2D23	FILX 299				P1=0\$K02	SET TRAP GATE BIT
2388	3D15	FILX 300				P1=P1\$K10	SET CONTROL BIT
238A	1D00	FILX 301	RSTWD			RST FIA K=80	RESET TRAP LATCH
238C	4BDF	FILX 302				TGRO=P1	SET TRAP GATE
238E	C222	FILX 303	RTNWD			RTN V MMSK1=0	RETURN FROM TRAP
23E4	CDE9	FILX 304	CETIME	306	WLRSET	BR IF FFIO=1	BR IF DATA CHAIN
23E6	F9EB	FILX 305		307	NOWLR	BR IF FF17=1	BR IF SLI
23E8	8E24	FILX 306	WLRSET	401	ICL	BR	
23EA	DD8F	FILX 307	NOWLR	303	RTNWD	BR IF FF11=1	BR IF COMMAND CHAIN
23EC	2E83	FILX 308				HO=0\$K08	SET CHANNEL END FOR STATUS
23EE	8E38	FILX 309		411	ZROSTA	BR	
2390	FFDF	FILX 310	EXCTLS	330	TSTOP4	BR IF FOP3=1	BR IF RECAL RESTORE OR SEEK HD
2392	CB07	FILX 311		333	P5TST	BR IF FOP4=1	BR IF SEEK CYL
2394	DBBA	FILX 312		338	NOOP	BR IF FOP5=0	BR IF NO OP
2396	C959	FILX 313	P4TST	334	PROTEC	BR IF P1 BIT4=1	FULL SEEK OR RECAL-TST FILE PROT
2398	FFE2	FILX 314		332	SKEX	BR IF FOP3=0	BR IF FULL SEEK
239A	2D07	FILX 315	RECAL			P1=0	
239C	4BDF	FILX 316				TGRO=P1	ZERO TAG REG
239E	2D23	FILX 317				P1=0\$K02	SET RTN TO 000 BIT
23A0	4EDF	FILX 318				FBO=P1	SET RTN TO 000 ON BUSS
23A2	2D1B	FILX 319	TMREC			P1=P1+K01	ADD TO GET TIME OUT
23A4	FOA2	FILX 320		319	TMREC	BR IF LZNZ	BR IF NOT TIMED OUT
23A6	4BDF	FILX 321				TGRO=P1	SET CONTROL TAG
23A8	5E32	FILX 322				RDH H DA,8E	GET STATS
23AA	1D00	FILX 323				RST FIA K=80	RESET TRAP LATCH
23AC	EE0E	FILX 324		303	RTNWD	BR IF HO BIT2=0	BR IF CAME FROM TRAP
23AE	2E85	FILX 325				HO=0\$K80	SET ACTIVE STAT
23B0	7E32	FILX 326				STH H DA,8E	STORE STATS
23B2	C3B7	FILX 327		329	IPLOP	BR IF BA4=1	BR IF IPL LATCH ON
23B4	B362	FILX 328		CCDM 154	CCOB	BR	SET CCO

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS	
23B6	94D8	FILX 329	IPL0P	FILT 097	IPL	BR	GO TO IPL WAIT LOOP	
23D8	CBB8	FILX 330	TSTOP4	337	TSTOP5	BR IF FOP4=0	BR IF RECAL OR RESTORE	
23E0	E959	FILX 331		334	PROTEC	BR IF P1 BIT6=1	SEEK HD TEST FOR FILE PROTECT	
23E2	9760	FILX 332	SKEX	201	SKSNS	BR		
23E6	D962	FILX 333	P5TST	332	SKEX	BR IF P1 BIT5=0	SEEK CYL TEST FOR FILE PROTECT	
23D8	2E25	FILX 334	PROTEC			HO=0\$K20	SET CONTINGENT CONNECTION BIT	
23DA	2F43	FILX 335				H1=0\$K04	SET FILE PROTECT BIT FOR SNS	
23DC	86BA	FILX 336		154	SETSEN	BR		
23B8	DB96	FILX 337	TSTOP5	313	P4TST	BR IF FOP5=0	BR IF RECALIBRATE	
23BA	5E32	FILX 338	NOOP			RDH H DA, 8E	GET STATS	
23BC	1E45	FILX 339				HO=HO*-K40	STRIP SID IND	
23BE	FFCE	FILX 340		346	ISNOOP	BR IF FOP3=0	BR IF NO OP	
23C0	F9CA	FILX 341	NOPSLI	350	TSTOP3	BR IF FFI7=0	BR IF NOT SLI	
23C2	CDCB	FILX 342		350	TSTOP3	BR IF FFI0=1	BR IF DATA CHAIN	
23C4	DDCA	FILX 343		350	TSTOP3	BR IF FFI1=0	BR IF NOT CMND CHAIN	
23C6	7E32	FILX 344	STRNOP			STH H DA, 8E	STORE STATS	
23C8	B238	FILX 345		114	CLDRST	BR	BR TO GET NEXT CCW	
23CE	F5C1	FILX 346	ISNOOP	341	NOPSLI	BR IF DASI3=1	BR IF ERASING	
23D0	CDCD	FILX 347		351	DCIL	BR IF FFI0=1	BR IF DATA CHAIN	
23D2	DDC7	FILX 348		344	STRNOP	BR IF FFI1=1	BR IF COMMAND CHAIN	
23D4	82C6	FILX 349		117	CEADE	BR		
23CA	7E32	FILX 350	TSTOP3			STH H DA, 8E	STORE STATS	
23CC	B22A	FILX 351	DCIL	107	COMEND	BR		
		FILX 352	*	*****				
		FILX 353	*				**	
		FILX 354	*				**	
		FILX 355	*			SEEK EXECUTE	**	
		FILX 356	*	*****				
		FILX 357	ASEQ	ALD7=4E				
0DCE	3F85	FILX 358	FWDSET			H1=H1\$K80	SET FORWARD INDICATOR	
0DD0	7E62	FILX 359				STH H DA, 9C	SAVE SEEK DATA	
0DD2	FFD6	FILX 360		362	GETCAR	BR IF FOP3=0	BR IF NOT HEAD SEEK	
0DD4	51EF	FILX 361	GETCYL			HO=FBI	SET CYL NUMBER FOR HD SEEK	
0DD6	51CF	FILX 362	GETCAR			PO=FBI	GET CYL ADDR TO COMPUTE DIFF	
0DD8	2DC7	FILX 363				P1=0		
0DDA	4BDF	FILX 364				TGRO=P1	RESET TAG REG	
0DDC	2D15	FILX 365				P1=0\$K10	CONTROL TAG&RST HEAD BIT	
0DDE	7CE3	FILX 366				PO=PO-HO	COMPUTE DIFFERENCE	
0DE0	F4E8	FILX 367		371	RSTHD	BR IF AC=0	BR IF DIR FWD-NO RECOMP NEEDED	
0DE2	2C1B	FILX 368				PO=PO+K01	ADD CARRY	
0DE4	1CFF	FILX 369				PO=PO+KFF	RECOMPLIMENT	
0DE6	5FFD	FILX 370				H1=H1L	REMOVE FORWARD BIT	
0DE8	4EDF	FILX 371	RSTHD			FBO=P1	PUT RESET HD ON BUSS	
0DEA	4BDF	FILX 372				TGRO=P1	SET CONTROL TAG	
0DEC	2DC7	FILX 373				P1=0		
0DEE	4BDF	FILX 374				TGRO=P1	RESET TAG REG	
0DF0	4EEF	FILX 375				FBO=HO	PUT CYL NO ON BUSS	
0DF2	2E45	FILX 376				HO=0\$K40	SET THE SET CYL BIT	
0DF4	4BEF	FILX 377				TGRO=HO	PUT SET CYL IN TAG REG	
0DF6	4BDF	FILX 378				TGRO=P1	RESET TAG REG	
0DF8	4EFF	FILX 379				FBO=H1	PUT HD NO ON BUSS	
0DFA	2E25	FILX 380				HO=0\$K20	SET THE SET HD BIT	
0DFC	4BEF	FILX 381				TGRO=HO	RAISE SET HD TAG	

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0DFE	4BDF	FILX 382				TGRD=P1	RESET TAG REG
0E00	0CFF	FILX 383				Z=POBKFF	MASK FOR ZERO DIFF
0E02	C496	FILX 384		394	MOTICN	BR IF ZNZ	BR IF DIFF NOT ZERO-MOTION SEEK
0E04	2F43	FILX 385				H1=0\$K04	SET SELECT HD BIT
0E06	4EFF	FILX 386				FBO=H1	PUT SELECT HD ON BUSS
0E08	2EC3	FILX 387				H0=0\$K0C	SET CE+DE FOR DEVICE STATUS
0E0A	DFA6	FILX 388		402	ISSEEK	BR IF FOP1=0	BR IF NOT FROM HD SSITCH ROUTINE
0E0C	2D15	FILX 389	SETCTL			P1=0\$K10	SET CONTROL BIT
0E0E	4BDF	FILX 390				TGRD=P1	RAISE CONTROL TAG
0E10	5E42	FILX 391				RDH H DA,98	GET CCW ADDR
0E12	1D00	FILX 392				RST FIA K=80	RESET TRAP LATCH
0E14	0222	FILX 393				RTN V MMSK1=0	RETURN FROM TRAP
0E16	4ECF	FILX 394	MOTION			FBO=PO	PUT DIFFERENCE ON BUSS
0E18	2FE5	FILX 395				H1=0\$K80	SETV THE SET DIFF BIT
0E1A	4BFF	FILX 396				TGRD=H1	RASSE SET DIFF TAG
0E1C	4BDF	FILX 397				TGRD=P1	RESET TAG REG
0E1E	4EEF	FILX 398				FBO=H0	PUT SEEK START ON BUSS
0E20	2D13	FILX 399				P1=0\$K01	SET CTL PULSE BIT
0E22	4BDF	FILX 400				TGRD=P1	ISSUE CONTROL PULSE
0E24	2E83	FILX 401	ICL			H0=0\$K08	SET CHANNEL END FOR STATUS
0E26	2F45	FILX 402	ISSEEK			H1=0\$K40	PRESET INCORRECT LEN STATUS
0E28	CDBB	FILX 403		412	ZROSN2	BR IF FF10=1	BR IF DATA CHAIN
0E2A	F9B1	FILX 404		407	SLION	BR IF FF17=1	BR IF SLI IS ON
0E2C	4886	FILX 405				I=I	TST FOR CCW COUNT 0
0E2E	C4BA	FILX 406		412	ZROSN2	BR IF ZNZ	BR IF ICL
0E30	DDB8	FILX 407	SLION	411	ZRCSTA	BR IF FF11=0	BR IF NOT CMND CHAIN
0E32	DA36	FILX 408		410	SETA	BR IF H0 BIT5=0	BR IF SEEK HAD MOTION
0E34	B236	FILX 409		113	NTORST	BR	GET NEXT CCW
0E36	86F4	FILX 410	SETA	183	GET8E	BR	
0E38	2FC7	FILX 411	ZROSTA			H1=0	ZERO CHAN STATUD
0E3A	2AC7	FILX 412	ZROSN2			T0=0	
0E3C	2B07	FILX 413				T1=0	
0E3E	7A62	FILX 414				STH T DA,9C	ZERO SENSE BYTE 2
0E40	1E00	FILX 415	CHNEND			RST FIB K=80	CHAIN END RESET
0E42	1D00	FILX 416				RST FIA K=80	RESET TRAP LATCH
0E44	2E08	FILX 417				SET FIB K=40	RESET FLAG BITS 6,7
0E46	5C32	FILX 418				RDH P DA,8E	GET STATS
0E48	C3DC	FILX 419		429	NOIPL	BR IF BA4=0	BR IF NOY IPL
0E4A	0EC3	FILX 420				Z=H0*-K0C	MASK FOR CORRECT DEVICE STATUS
0E4C	F0D1	FILX 421		423	CHANTS	BR IF LZ=0	BR IF DEVICE STATUA OK
0E4E	5007	FILX 422	STOP			STOP	IPL STOP WORD
0E50	0F85	FILX 423	CHANTS			Z=H1*-K80	MASK FOR GOOD CHAN STATUS
0E52	C4CE	FILX 424		422	STOP	BR IF ZNZ	BR IF CHAN STATUA BAD
0E54	2C07	FILX 425				PO=0	REMOVE ACTIVE STAT
0E56	7C32	FILX 426				STH P DA,8E	
0E58	0212	FILX 427				RST MMSK K=11	RESET PRIORITY
0E5A	8776	FILX 428		DCLL 036	PICKUP	BR	BR TO LOAD PSW
0E5C	2B43	FILX 429	NOIPL			T1=0\$K04	SET AUX ADDR 0004
0E5E	7CA8	FILX 430				STH P AS,T+2	STORE I/O ADDR FOR CONT CONN
0E60	DC6E	FILX 431		438	STRINT	BR IF PO BIT1=0	BR IF NOT FROM SID
0E62	7A32	FILX 432				STH T DA,8E	RESET STATS
0E64	C9EA	FILX 433		436	NOPCI	BR IF FF14=0	BR IF PCI OFF
0E66	3F85	FILX 434				H1=H1\$K80	SET PCI IN ST&TUS

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
0E68	0D04	FILX 435				RST FIA K=20	RESET PCI LATCH
0E6A	2410	FILX 436	NUPCI			SET MODE K=01	SET CPU MODE FILE ZONE
0E6C	A030	FILX 437		CCOM 135	STATUS	BR	SET CCI
0E6E	7EA0	FILX 438	STRINT			STH H AS,T+0	STORE STATUS IN INTERRUPT BUFF
0E70	2D08	FILX 439				SET FIA K=40	SET INTERRUPT LATCH
0E72	4DAF	FILX 440				FFO=TO	RESET FLAG REG
0E74	4FAF	FILX 441				FDP=TO	RESET OP REG
0E76	EC7E	FILX 442		446	NORIC	BR IF P0 BIT2=0	BR IF NOT RETURN TO I CYCLES
0E78	2CE5	FILX 443				PU=0\$K80	SET ACTIVE STAT
0E7A	7C32	FILX 444				STH P DA,8E	STORE STATS
0E7C	B362	FILX 445		CCOM 154	CCOB	BR	SET CCO
0E7E	0222	FILX 446	NORIC			RTN V MMSK1=0	RETURN FROM TRAP
		FILX 447	AEND				

 * CROSS REFERENCE FOR CSECT FILX *

FILX 007	FILT 042						
FILX 015	FILX 022	FILX 023	FILX 050	FILX 051	FILX 057	FILX 069	
FILX 028	FILX 030						
FILX 029	FILX 025						
FILX 038	FILX 027	FILX 029					
FILX 040	FILX 031						
FILX 042	FILX 042						
FILX 049	FILX 009						
FILX 053	FILX 053						
FILX 057	FILX 017						
FILX 058	FILX 049	FILX 060	FILX 061	FILX 062			
FILX 059	FILX 013						
FILX 060	FILX 021						
FILX 061	FILX 059						
FILX 069	FILX 010	FILX 014	FILX 018				
FILX 079	FILX 041						
FILX 082	FILX 084						
FILX 083	FILX 080						
FILX 086	FILX 252						
FILX 088	FILT 055	FILX 082					
FILX 091	FILX 089						
FILX 092	FILX 090						
FILX 097	FILX 094						
FILX 100	FILX 095	FILX 097					
FILX 104	FILX 100						
FILX 106	FILX 043	FILX 102	FILX 104				
FILX 107	FILX 237	FILX 243	FILX 351				
FILX 112	FILX 109						
FILX 113	FILT 095	FILX 409					
FILX 114	FILX 345						
FILX 117	FILX 112	FILX 349					
FILX 118	FILX 108	FILX 111	FILX 247				
FILX 125	FILX 012						
FILX 129	FILX 125						
FILX 132	FILX 126						
FILX 134	FILX 138						
FILX 135	FILX 063	FILX 164					

 * CROSS REFERENCE FOR CSECT FILX *

FILX 298	FILX 073		
FILX 303	FILX 307	FILX 324	
FILX 304	FILX 296		
FILX 306	FILX 304		
FILX 307	FILX 305		
FILX 310	FILX 298		
FILX 313	FILX 337		
FILX 315	FILX 056		
FILX 319	FILX 320		
FILX 329	FILX 327		
FILX 330	FILX 310		
FILX 332	FILX 314	FILX 333	
FILX 333	FILX 311		
FILX 334	FILT 141	FILX 313	FILX 331
FILX 337	FILX 330		
FILX 338	FILX 312		
FILX 341	FILX 346		
FILX 344	FILX 348		
FILX 346	FILX 340		
FILX 350	FILX 341	FILX 342	FILX 343
FILX 351	FILX 347		
FILX 358	FILX 289		
FILX 361	FILT 140		
FILX 362	FILX 360		
FILX 371	FILX 367		
FILX 389	FILT 128		
FILX 394	FILX 384		
FILX 401	FILX 306		
FILX 402	FILX 388		
FILX 407	FILX 404		
FILX 410	FILX 408		
FILX 411	FILX 156	FILX 309	FILX 407
FILX 412	FILT 066	FILX 119	FILX 260
FILX 415	FILT 197	FILX 403	FILX 406
FILX 422	FILX 424		
FILX 423	FILX 421		
FILX 429	FILX 419		
FILX 436	FILX 433		
FILX 438	FILX 431		
FILX 446	FILX 442		

FINT DESCRIPTIVE TEXT

ENTRY POINTS

INTRPT THIS IS THE NORMAL ENTRY INTO THE ROUTINE WHEN A FILE INTERRUPT IS DETECTED IN THE BSWI ROUTINE. THIS ROUTINE TESTS FOR FILE CONDITIONS BEFORE SETTING THE CSW INFORMATION INTO THE APPROPRIATE REGISTERS FOR THE CCOM ROUTINE.

SETCSW THIS IS A BAL ENTRY TO SET THE CSW INFORMATION INTO THE CPU REGISTERS FOR STORING BY THE CCOM ROUTINE.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		FINT 001	T			2311 FEATURE INTERUPT ROUTINE FOR FILES	
25FC	7CC2	FINT 002	INTRPT			STH P DA,88	SAVE CPU P REG
25FE	5F59	FINT 003				G1=H1	SAVE CPU H1 REG
2600	D58E	FINT 004		011	TSTPCI	BR IF FF15=0	BR IF NOT INETRUP T LATCH
2602	5A32	FINT 005				RDH T DA,8E	GET STATS AND IOA
2604	2A13	FINT 006				T0=0\$K01	RESET STATS
2606	7A32	FINT 007				STH T DA,8E	STORE STATS AND IOA
2608	8420	FINT 008		052	SETCSW	BAL	SET REGS FOR CPU CSW STORE
260A	A024	FINT 009	STUFCS	CCOM	127	CSWADR	BAL
260C	A64C	FINT 010		042	GETIOA	BR	STORE CSW
260E	C996	FINT 011	TSTPCI	015	SETIOA	BR IF FF14=0	BR IF NOT PCI
2610	2E07	FINT 012				H0=0	SET DEVICE STATUS
2612	8428	FINT 013		056	PCICSW	BAL	SET CHAN STATUS PCI INFO
2614	A60A	FINT 014		009	STUFCS	BR	
2616	2B43	FINT 015	SETIOA			T1=0\$K04	SET AUX ADDR 0004
2618	4FBF	FINT 016				FQP=T1	SET CONTIN CONECT CLEAR IND
261A	5AA0	FINT 017				RDH T AS,T+0	GET BUFFERED IO ADDR FROM 0004
261C	2A13	FINT 018				T0=0\$K01	RESET STATS
261E	7A32	FINT 019				STH T DA,8E	STORE STATS&I/O ADDR
2620	E1A8	FINT 020		024	TSTDE	BR IF DAS16=0	BR IF NOT CUE
2622	0D02	FINT 021				RST FIA K=10	RESET CUB LATCH
2624	2E25	FINT 022				H0=0\$K20	SET CUE FOR STATUS
2626	A644	FINT 023		038	STOSTA	BR	
2628	E5AF	FINT 024	TSTDE	027	RSTDE	BR IF DAS12=1	BR IF GATTED ATTENTION
262A	948E	FINT 025		FILE	264	SNSSTA	BAL
262C	A644	FINT 026		038	STOSTA	BR	ERROR-NO REASON FOR INT-SET SNS
262E	AFEA	FINT 027	RSTDE	FILE	280	SETDA	BAL
2630	5DDD	FINT 028				P1=P1L	SET DEV ADDR&SELECT MODULE
2632	58BB	FINT 029				T1=T1H	BUILD
2634	6BD5	FINT 030				T1=T1\$P1	DEVICE ADDRESS
2636	7A32	FINT 031				STH T DA,8E	AND
2638	FABF	FINT 032		035	SEKINC	BR IF DS7=1	STORE IT IN AUX
263A	8224	FINT 033		FILE	319	RSTATT	BAL
263C	A642	FINT 034		037	SETDE	BR	BR IF SEEK INCOMPLETE
263E	8224	FINT 035	SEKINC	FILE	319	RSTATT	BAL
							RESET GATED ATTN

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS	
2640	9456	FINT 036		FILE 268	SNSSTC	BAL	SET SEEK CHECK SNS	
2642	3E43	FINT 037	SETDE			H0=H0\$K04	SET DE FOR STATUS	
2644	2F07	FINT 038	STOSTA			H1=0	ZERO CHAN STATUS	
2646	2400	FINT 039	MODE			SET MODE K=00	SET CPU ZONE&MODE	
2648	2AC7	FINT 040				T0=0		
264A	A01E	FINT 041		CCOM 123	STATOS	BAL	ZERO CSW THEN STORE STATUS	
264C	5A32	FINT 042	GETIOA			RDH T DA,8E	GET I/O ADDRESS	
264E	2A13	FINT 043				T0=0\$K01	SET CHAN 1	
2650	1000	FINT 044				RST S0		
2652	5CC2	FINT 045				RDH P DA,B8	RESTORE P REG	
2654	A43C	FINT 046		DCHN 074	FRMFIL	BK		
		FINT 047	*	*****				
		FINT 048	*	*				**
		FINT 049	*			SET CSW	- SUBROUTINE	
		FINT 050	*	*				**
		FINT 051	*	*****				
0420	2AC7	FINT 052	SETCSW			T0=0		
0422	2B63	FINT 053				T1=0\$K06	SET AUX ADDR 0006	
0424	5EA0	FINT 054				RDH H AS,T+0	GET BUFFERED STATUS	
0426	0D08	FINT 055				RST FIA K=40	RESET INTURRUPT LATCH	
0428	526F	FINT 056	PCICSW			DO=STP0	GET PROTECT KEY	
042A	566B	FINT 057				DO=DOH		
042C	2707	FINT 058				D1=0		
042E	C9B2	FINT 059		061	NOPCI	BR IF FFI4=0	BR IF NOT PCI INT	
0430	3F85	FINT 060				H1=H1\$K80	SET PCI IN STATUS	
0432	2490	FINT 061	NOPCI			SET MODE K=09	SET FILE MODE,FILE ZONE	
0434	78A2	FINT 062				STH I DA,AC	BUFFER RESIDUAL COUNT	
0436	0D04	FINT 063				RST FIA K=20	RESET PCI LATCH	
0438	2400	FINT 064				SET MODE K=00	SET CPU MODE AND ZONE	
043A	50A2	FINT 065				RDH U DA,AC	GET BUFFERED RESID COUNT	
043C	5242	FINT 066				RDH V DA,98	GET CCW NEXT ADDR	
043E	128E	FINT 067				RTN		

 * CROSS REFERENCE FOR CSECT FINT *

FINT 002	DCLT 052	
FINT 009	FINT 014	
FINT 011	FINT 0C4	
FINT 015	FINT 011	
FINT 024	FINT 020	
FINT 027	FINT 024	
FINT 035	FINT 032	
FINT 037	FINT 034	
FINT 038	FINT 023	FINT 026
FINT 042	FINT 010	
FINT 052	FILE 017	FINT 008
FINT 056	FINT 013	
FINT 061	FINT 059	

ADDRESS LIST

DATE 11/08/68

03--

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-0300-	CMLT 018 29F7	CMLT 019 2C13	CMLT 020 0275	CMLT 021 E08B	CMLT 022 3C23	CMLT 023 5989	CMLT 024 4A86	CMLT 025 00C0	-0300-
-0310-	CMLT 026 7CF2	CMLT 027 4C52	CMLT 028 2D1D	CMLT 029 2D8B	CMLT 030 CD29	CMLT 031 78C8	CMLT 032 7AC8	CMLT 033 76C8	-0310-
-0320-	CMLT 034 7EC8	CMLT 035 856E	CMLT 036 856E	CMLT 037 8314	CMLT 038 2D07	CMLT 039 48C6	CMLT 040 4AC6	CMLT 041 44C6	-0320-
-0330-	CMLT 042 46C6	CMLT 043 4E06	CMLT 044 40C6	CMLT 045 2C13	CMLT 046 C5BE	CMLT 047 2F1B	CMLT 048 F4D3	CMLT 049 2D15	-0330-
-0340-	CMLT 050 1002	CMLT 051 8B55	CMLT 052 DD53	CMLT 053 C5CC	CMLT 054 2F1D	CMLT 055 F4D3	CMLT 056 2D55	CMLT 057 1002	-0340-
-0350-	CMLT 058 BF55	CMLT 059 2C2B	CMLT 060 9468	CMLT 061 C838	CMLT 062 C1DC	CMLT 063 95F6	CMLT 064 54F2	CMLT 065 C5E9	-0350-
-0360-	CMLT 066 E068	CMLT 067 3002	CMLT 068 2D15	CMLT 069 9CD2	CMLT 070 9256	CMLT 071 2A07	CMLT 072 558B	CMLT 073 72A8	-0360-
-0370-	CMLT 074 7EAA	CMLT 075 F47A	CMLT 076 1B15	CMLT 077 76A8	CMLT 078 70AA	CMLT 079 A8DE	CCOM 029 55B3	CCOM 030 56A8	-0370-
038-									
-0380-	CCOM 031 5EAA	CCOM 032 558B	CCOM 033 0008	CCOM 034 128E	BCHK 098 07FF	BCHK 098 C48A	BCHK 098 8800	BCHK 098 4E08	-0380-
-0390-	CMLT 177 3002	CMLT 178 9CD2	CMLT 179 3002	CMLT 180 80F4	BCHK 101 00C8	BCHK 101 0000	BCHK 104 4E08	BCHK 104 128E	-0390-
-03A0-	DYPE 226 8DC4	DYPE 227 A568	DYPE 228 9028	DYPE 229 9F20	BCHK 108 0000	BCHK 108 0000	BCHK 108 0000	BCHK 108 0000	-03A0-
-03B0-	DCLR 196 2343	DCLR 197 5612	DCLR 198 3613	DCLR 199 A29C	BCHK 123 2FF7	BCHK 123 2F1B	BCHK 123 C4BC	BCHK 123 0F01	-03B0-
-03C0-	BDIA 347 C4C0	BDIA 348 4AC2	BDIA 349 4CD2	BDIA 350 4EF2	BDIA 351 2A1B	BDIA 352 2C1B	BDIA 353 2E1B	BDIA 354 E0F2	-03C0-
-03D0-	BDIA 355 6AC2	BDIA 356 6CD2	BDIA 357 6EF2	BDIA 358 8388	BDIA 331 4812	BDIA 332 3935	BDIA 333 292B	BDIA 334 68E2	-03D0-
-03E0-	BDIA 335 2A25	BDIA 336 2BF5	BDIA 337 3873	BDIA 338 2C25	BDIA 339 2D15	BDIA 340 3DB3	BDIA 341 2E05	BDIA 342 2F13	-03E0-
-03F0-	BDIA 343 83D0	BDIA 373 2505	BDIA 374 251B	BDIA 375 F4FD	BDIA 376 C4F4	BDIA 377 5007	BDIA 381 2413	BDIA 382 883C	-03F0-

CONTROL ADDRESS

03--

ADDRESS LIST		DATE 11/08/68							
CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-0400-	CICY 203 AEE8	CICY 204 ACB0	CICY 205 A93A	CICY 206 AE6C	CICY 207 AE22	CICY 208 AC7A	CICY 209 ABF6	CICY 210 AD72	-0400-
-0410-	CICY 211 9AB0	CICY 217 A05A	CICY 219 A05A	CICY 220 A05A	CICY 221 8CB8	CICY 222 8CB8	CICY 223 8CB8	CICY 224 8CB8	-0410-
-0420-	FINT 052 2A07	FINT 053 2B63	FINT 054 5EA0	FINT 055 0D08	FINT 056 526F	FINT 057 566B	FINT 058 2707	FINT 059 C9B2	-0420-
-0430-	FINT 060 3F85	FINT 061 2490	FINT 062 78A2	FINT 063 0D04	FINT 064 2400	FINT 065 50A2	FINT 066 5242	FINT 067 128E	-0430-
-0440-	BDIA 122 5007	DCLB 006 2045	DCLB 007 7022	DCLB 009 2625	DCLB 010 2787	DCLB 011 81F8	DCLB 013 2486	DCLB 014 2880	-0440-
-0450-	DCLB 015 7212	DCLB 016 24F6	DCLB 017 5622	DCLB 018 2407	DCLB 019 2585	DCLB 020 3543	DCLB 021 5148	DCLB 022 6171	-0450-
-0460-	DCLB 023 C4F1	DCLB 024 C15C	DCLB 025 25CB	DCLB 026 F55D	DCLB 027 2507	DCLB 028 8472	DCLB 004 C642	DCLB 005 AC74	-0460-
-0470-	DCLB 029 5651	DCLB 030 5212	DCLB 031 93F4	DCLR 106 3425	DCLR 107 9616	FILE 008 7CC2	FILE 009 2480	FILE 010 C628	-0470-
048-									
-0480-	FILE 011 E011	FILE 012 F01E	FILE 013 D99E	FILE 014 6711	FILE 015 C49E	FILE 016 7032	FILE 017 8420	FILE 018 A024	-0480-
-0490-	FILE 019 5DBF	FILE 020 1BC5	FILE 021 D9A1	FILE 022 3210	FILE 023 2D04	FILE 024 4DBF	FILE 025 1210	FILE 026 AC74	-0490-
-04A0-	FILE 027 F5BB	FILE 028 2E07	FILE 029 D9BE	FILE 030 B362	FILE 031 3065	FILE 032 7032	FILE 033 E021	FILE 034 F5BB	-04A0-
-04B0-	FILE 035 E1C1	FILE 036 F054	FILE 037 DBFD	FILE 038 8702	FILE 039 DBFC	FILE 040 2E55	FILE 041 2D02	FILE 042 854E	-04B0-
-04C0-	FILE 043 DBCA	FILE 044 8702	FILE 045 DBBB	FILE 046 2E43	FILE 047 4FEF	FILE 048 2E25	FILE 049 F051	FILE 050 2E35	-04C0-
-04D0-	FILE 051 0D02	FILE 052 854E	FILE 053 7242	FILE 054 5AF2	FILE 055 42AF	FILE 056 5A30	FILE 057 0A4B	FILE 058 C4E4	-04D0-
-04E0-	FILE 059 AFC6	FILE 060 8C92	FILE 061 DBEB	FILE 062 8702	FILE 063 DBBB	FILE 064 0A3B	FILE 065 C4FD	FILE 066 0A1D	-04E0-
-04F0-	FILE 067 E0F6	FILE 068 0A7B	FILE 069 F0FD	FILE 070 0133	FILE 071 F0FD	FILE 072 850C	FILE 073 5ECF	FILE 074 C487	-04F0-

CONTROL ADDRESS

04--

ADDRESS LIST

DATE 11/08/68

OB--

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-0B00-	DCLC 023 C9DD	DCLC 024 FDA1	DCLC 025 262B	DCLC 026 E600	DCLC 027 05C3	DCLC 028 F094	DCLC 029 F514	DCLC 030 2323	-0B00-
-0B10-	DCLC 031 2207	DCLC 032 8B3A	DCLC 033 0D9F	DCLC 034 2B00	DCLC 035 F21F	DCLC 036 C3CB	DCLC 037 AAF4	DCLC 038 A254	-0B10-
-0B20-	DCLC 039 5F4F	DCLC 040 FC9F	DCLC 041 2B84	DCLC 042 E61F	DCLC 043 FDAE	DCLC 044 262B	DCLC 045 FDA7	DCLC 046 2B80	-0B20-
-0B30-	DCLC 047 F21F	DCLC 048 E149	DCLC 049 F149	DCLC 050 5429	DCLC 051 2307	DCLC 052 72A2	DCLC 053 2B00	DCLC 054 2400	-0B30-
-0B40-	DCLC 055 5EA2	DCLC 056 2A07	DCLC 057 A01E	DCLC 058 AF44	DCLC 059 C3D8	DCLC 060 875A	DCLC 061 F537	DCLC 062 5449	-0B40-
-0B50-	DCLC 063 C4B6	DCLC 064 2B00	DCLC 065 B362	DCLC 066 2B84	DCLC 067 2307	DCLC 068 A78A	DCLC 069 2BC0	DCLC 070 0D10	-0B50-
-0B60-	DCLC 071 E61F	DCLC 072 CDE9	DCLC 073 262B	DCLC 074 CDE0	DCLC 075 4F4F	DCLC 076 5F4F	DCLC 077 FC9F	DCLC 078 6471	-0B60-
-0B70-	DCLC 079 C49E	DCLC 080 054B	DCLC 081 C4D7	DCLC 082 2BC2	DCLC 083 9E06	CICY 013 5892	CICY 014 2020	CICY 015 87C8	-0B70-
									0B8-
-0B80-	CDMD 150 A520	CDMD 106 571A	CDMD 107 A317	CDMD 108 2FFF	CDMD 109 F483	CDMD 110 5004	CDMD 111 5224	CDMD 112 5EF9	-0B80-
-0B90-	CDMD 113 2B07	CDMD 114 E597	CDMD 115 2040	CDMD 116 0004	CDMD 117 77AF	CDMD 118 5730	CDMD 119 D1A1	CDMD 120 7718	-0B90-
-0BA0-	CDMD 121 7B38	CDMD 122 2FFF	CDMD 123 C498	CDMD 124 5006	CDMD 125 5226	CDMD 126 D1C3	CDMD 127 577B	CDMD 128 7710	-0BA0-
-0BB0-	CDMD 129 77AF	CDMD 130 C181	CDMD 131 E5C2	CDMD 132 B1BC	CDMD 133 0B9D	CDMD 134 EOC2	CDMD 135 4086	CDMD 136 1002	-0BB0-
-0BC0-	CDMD 137 9AF0	CDMD 138 54F9	CDMD 139 9254	CDMD 140 27C3	CDMD 141 C94C	CDMD 142 176B	CDMD 143 EF51	CDMD 144 3713	-0BC0-
-0BD0-	CDMD 145 5F10	CDMD 146 6F75	CDMD 147 7F10	CDMD 148 2A07	CDMD 149 A8DE	CFCO 056 0008	CFCO 057 6FF5	CFCO 058 4886	-0BD0-
-0BE0-	CFCO 059 F467	CFCO 060 4226	CFCO 061 4006	CFCO 062 128E	FILE 137 8114	FILE 138 3E00	FILE 139 2490	FILE 140 0E08	-0BE0-
-0BF0-	FILE 141 2302	FILE 142 2304	FILE 143 3110	FILE 144 3D29	FILE 145 4FDF	FILE 146 43DF	FILE 147 B344	FILE 148 5FCF	-0BF0-

CONTROL ADDRESS

OB--

ADDRESS LIST		DATE 11/08/68							
CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1000-	CICY 195 ABC2	CICY 196 A05A	CICY 197 AB7E	CICY 198 AB8A	CICY 199 8302	CICY 200 830C	CICY 201 ACC8	CICY 202 ACC8	-1000-
-1010-	CICY 050 9BF4	CICY 047 9A6A	CICY 048 C011	CICY 049 9001	BWRP 066 2A07	BWRP 067 2843	BWRP 068 2F07	BWRP 069 9C72	-1010-
-1020-	DCLB 209 9CFC	DCLB 211 9812	DCLB 212 9812	DCLB 214 A4A2	DYPE 168 2791	DYPE 169 566D	DYPE 170 16E3	DYPE 171 2404	-1020-
-1030-	DYPE 172 5C68	DYPE 173 5468	DYPE 174 5268	DYPE 175 5068	DYPE 176 52E9	DYPE 177 2771	DYPE 178 128E	5007	-1030-
-1040-	DCLB 199 8AEE	DPTC 067 0C3B	DPTC 068 F0CA	DPTC 070 8616	DPTC 081 8612	DPTC 074 0CBB	DPTC 075 F0C8	DPTC 077 8604	-1040-
-1050-	EXFR 007 2455	EXFR 008 2515	EXFR 009 3593	EXFR 010 2045	EXFR 011 21A5	EXFR 012 22A3	EXFR 013 2363	EXFR 014 4852	-1050-
-1060-	EXFR 015 4A86	EXFR 016 4C86	EXFR 017 4E86	EXFR 018 2613	EXFR 019 5700	EXFR 020 C76F	EXFR 021 6965	EXFR 022 D773	-1060-
-1070-	EXFR 023 6B65	EXFR 024 E777	EXFR 025 6D65	EXFR 026 F77B	EXFR 027 6F65	EXFR 028 C37F	EXFR 029 6865	EXFR 030 D303	-1070-
108-									
-1080-	EXFR 031 6A65	EXFR 032 E307	EXFR 033 6C65	EXFR 034 F30B	EXFR 035 6E65	EXFR 036 6663	EXFR 037 23FF	EXFR 038 C49F	-1080-
-1090-	EXFR 040 F41A	EXFR 042 21AB	EXFR 043 F498	EXFR 044 201D	EXFR 045 9068	EXFR 047 319B	EXFR 048 9068	EXFR 050 7940	-1090-
-10A0-	EXFR 051 6444	EXFR 052 7B40	EXFR 053 6444	EXFR 054 7D40	EXFR 055 6444	EXFR 056 7F40	EXFR 057 6444	EXFR 058 144B	-10A0-
-10B0-	EXFR 059 D03C	EXFR 060 5899	EXFR 061 5AB9	EXFR 062 5CD9	EXFR 063 5EF9	EXFR 064 909E	EXFR 065 22FF	EXFR 066 C4CF	-10B0-
-10C0-	EXFR 068 F44A	EXFR 070 313D	EXFR 071 C048	EXFR 072 20FD	EXFR 073 905C	EXFR 075 213F	EXFR 076 905C	EXFR 078 148B	-10C0-
-10D0-	EXFR 079 C060	EXFR 081 F45A	EXFR 083 211B	EXFR 084 35AD	EXFR 085 905A	EXFR 087 21A7	EXFR 088 2507	EXFR 089 905A	-10D0-
-10E0-	EXFR 090 F475	EXFR 091 52E2	EXFR 092 A062	EXFR 093 51A2	EXFR 094 A044	EXFR 095 54B2	EXFR 096 3659	EXFR 097 2845	-10E0-
-10F0-	EXFR 098 2907	EXFR 099 98DE	EXFR 100 2466	EXFR 104 E67E	EXFR 105 1625	EXFR 106 2400	EXFR 107 AAA2	EXFR 108 2406	-10F0-
CONTROL ADDRESS									10--

ADDRESS LIST

DATE 11/08/68

11--

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1100-	EXFR 109 A06A	EXFR 110 A044	EXFR 111 5FB2	EXFR 112 F80A	EXFR 113 2643	EXFR 114 FD8E	EXFR 115 3D15	EXFR 116 C992	-1100-
-1110-	EXFR 117 3D83	EXFR 118 2E23	EXFR 119 3485	EXFR 120 A923	EXFR 121 2E07	EXFR 122 F822	EXFR 123 E523	EXFR 124 3445	-1110-
-1120-	EXFR 125 1545	EXFR 126 8964	EXFR 127 0529	EXFR 128 3E83	EXFR 129 022E	EXFR 130 56F2	EXFR 131 8D58	EXFR 132 51A2	-1120-
-1130-	EXFR 133 2007	EXFR 134 D83A	EXFR 135 52F2	EXFR 136 613B	EXFR 137 602D	EXFR 138 C844	EXFR 139 3E43	EXFR 140 C3C4	-1130-
-1140-	EXFR 141 2820	EXFR 142 9D54	EXFR 143 C3C3	EXFR 144 8D44	CBIN 005 8A70	CBIN 006 A8CA	CSAS 006 3443	CSAS 007 1082	-1140-
-1150-	CSAS 008 2E15	CSAS 009 573A	CSAS 010 576D	CSAS 011 0661	CSAS 012 E0F3	CSAS 013 00C4	CSAS 014 F366	CSAS 015 07FB	-1150-
-1160-	CSAS 016 F0E7	CSAS 017 C5F5	CSAS 018 3042	CSAS 019 C07C	CSAS 020 D07B	CSAS 021 148B	CSAS 022 57FB	CSAS 023 5710	-1160-
-1170-	CSAS 024 9154	CSAS 025 807C	CSAS 026 1002	CSAS 027 2040	CSAS 028 9166	CSAS 029 1002	CSAS 030 577B	CSAS 031 77FF	-1170-
118-									
-1180-	CSAS 032 C905	CSAS 033 272B	CSAS 034 D188	CSAS 035 271B	CSAS 036 27AB	CSAS 037 F035	CSAS 038 771A	CSAS 039 55B9	-1180-
-1190-	CSAS 040 E0EB	CSAS 041 FOA6	CSAS 042 2707	CSAS 043 2EF3	CSAS 044 D02A	CSAS 045 2F07	CSAS 046 91AC	CSAS 047 7F1A	-1190-
-11A0-	CSAS 048 75E3	CSAS 049 E0EB	CSAS 050 F095	CSAS 051 573A	CSAS 052 D01B	CSAS 053 5F10	CSAS 054 7F7F	CSAS 055 F01E	-11A0-
-11B0-	CSAS 056 5006	CSAS 057 91A0	CSAS 058 5006	CSAS 059 918E	CSAS 060 1413	CSAS 061 3002	CSAS 062 9150	CSAS 063 573A	-11B0-
-11C0-	CSAS 064 F5D1	CSAS 065 7A7F	CSAS 066 1A9F	CSAS 067 C4F9	CSAS 068 0002	CSAS 069 1415	CSAS 070 2A07	CSAS 071 91F8	-11C0-
-11D0-	CSAS 072 E5D9	CSAS 073 7A7F	CSAS 074 C4F9	CSAS 075 91CA	CSAS 076 7A7F	CSAS 077 F5E1	CSAS 078 E445	CSAS 079 91CA	-11D0-
-11E0-	CSAS 080 142D	CSAS 081 91F8	CSAS 082 77AF	CSAS 083 91F6	CSAS 084 9172	CSAS 085 F0FD	CSAS 086 C5BF	CSAS 087 573A	-11E0-
-11F0-	CSAS 088 F5E5	CSAS 089 77AF	CSAS 090 C4F9	CSAS 091 2002	CSAS 092 25FF	CSAS 093 F0EC	CSAS 094 C1E9	CSAS 095 C591	-11F0-

CONTROL ADDRESS

11--

ADDRESS LIST

DATE 11/08/68

12--

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1200-	CSAS 096 F589	CSAS 097 E58B	CSAS 098 D1D1	CSAS 099 A8DC	CSAS 100 F03C	CSAS 101 D18F	CSAS 102 ACA4	CSAS 103 AC84	-1200-
-1210-	CSAS 104 F583	CSAS 105 F047	CSAS 106 5B85	CSAS 107 6189	CSAS 108 60AD	CSAS 109 5710	CSAS 110 57FB	CSAS 111 3002	-1210-
-1220-	CSAS 112 577D	CSAS 113 171B	CSAS 114 7AFF	CSAS 115 67A3	CSAS 116 771A	CSAS 117 5BA9	CSAS 118 C48B	CSAS 119 5710	-1220-
-1230-	CSAS 120 757F	CSAS 121 751A	CSAS 122 2507	CSAS 123 2AFF	CSAS 124 C4AE	CSAS 125 F447	CSAS 126 2C75	CSAS 127 DD43	-1230-
-1240-	CSAS 128 A8DE	CSAS 129 28A3	CSAS 130 9C70	CSAS 131 D1CD	CSAS 132 2040	CSAS 133 9202	CSAS 134 0040	CSAS 135 9202	-1240-
-1250-	CSAS 136 F007	CSAS 137 9214	CCOM 064 50A2	CCOM 065 54B2	CCOM 066 5CC2	CCOM 067 5892	CCOM 068 2020	CCOM 069 128E	-1250-
-1260-	BWRP 086 9D52	BWRP 087 809E	BWRP 088 809E	BWRP 089 9F20	BWRP 073 2607	BWRP 074 2E95	BWRP 075 C4F2	BWRP 076 2607	-1260-
-1270-	BWRP 077 2EA5	BWRP 078 1773	BWRP 079 5460	BWRP 080 64E5	BWRP 081 2543	BWRP 082 2EC3	BWRP 083 7468	BWRP 084 7E6A	-1270-
128-									
-1280-	BWRP 085 8765	DCLE 071 0313	DCLE 072 C486	DCLE 073 9E54	DCLE 046 5F7F	DCLE 047 FC9D	DCLE 048 2007	DCLE 049 4F0F	-1280-
-1290-	DCLE 050 2802	DCLE 051 E61D	DCLE 052 FD9B	DCLE 053 262B	DCLE 054 FD92	DCLE 055 A7B0	DCLE 045 A254	DCLE 058 FCA4	-1290-
-12A0-	DCLE 059 0D02	DCLE 060 3483	DCLE 061 5439	DCLE 062 C52A	DCLE 063 1565	DCLE 064 5F4F	DCLE 065 FC9D	DCLE 066 E503	-12A0-
-12B0-	DCLE 067 0583	DCLE 068 F083	DCLE 069 3345	DCLE 070 A782	DCLE 038 CD89	DCLE 039 262B	DCLE 040 CD89	DCLE 041 E638	-12B0-
-12C0-	DCLE 042 DD9D	DCLE 043 2B00	DCLE 044 0244	5007	EPCH 034 2462	EPCH 035 3643	EPCH 036 2402	EPCH 037 1C83	-12C0-
-12D0-	EPCH 041 930C	EPCH 032 98B4	EPCH 010 1F00	EPCH 014 FA74	EPCH 016 6CF7	EPCH 017 EA53	EPCH 018 CE62	EPCH 019 DE53	-12D0-
-12E0-	EPCH 020 3C45	EPCH 021 DE66	EPCH 022 3C25	EPCH 023 E853	EPCH 024 CA48	EPCH 026 3C23	EPCH 027 2F40	EPCH 028 2462	-12E0-
-12F0-	EPCH 029 1643	EPCH 030 92CC	EPCH 043 EA52	EPCH 044 DA53	EPCH 045 0E49	EPCH 046 F4D2	EPCH 047 EE53	EPCH 048 EF84	-12F0-

CONTROL ADDRESS

12--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1300-	EPCH 049 2C07	EPCH 050 98B0	EPCH 052 C808	EPCH 053 2D23	EPCH 054 3C83	EPCH 055 1C23	EPCH 056 3659	EPCH 057 5009	-1300-
-1310-	EPCH 058 C4A3	EPCH 059 3C53	EPCH 060 4A06	EPCH 061 2007	EPCH 062 21A5	EPCH 063 2002	EPCH 064 7B19	EPCH 065 7A09	-1310-
-1320-	EPCH 066 7AF2	EPCH 067 8964	EPCH 068 71A2	EPCH 069 72E2	EPCH 070 F72C	EPCH 071 98C4	EPCH 073 C82B	EPCH 074 CC42	-1320-
-1330-	EPCH 078 5CD9	EPCH 079 59C9	EPCH 080 C85E	EPCH 081 5809	EPCH 082 D83F	EPCH 083 5224	EPCH 084 282B	EPCH 085 58D5	-1330-
-1340-	EPCH 086 99B0	EPCH 087 2807	EPCH 088 2C45	EPCH 089 2DC3	EPCH 090 28F7	EPCH 091 5899	EPCH 092 54A0	EPCH 093 78A8	-1340-
-1350-	EPCH 094 74C8	EPCH 095 CD4C	EPCH 096 D94C	EPCH 097 5884	EPCH 098 3CE9	EPCH 099 2D07	EPCH 100 EE63	EPCH 101 1CA3	-1350-
-1360-	EPCH 102 9B54	EPCH 103 994A	CFCO 103 F46F	CFCO 104 711B	CFCO 105 700B	CFCO 106 733B	CFCO 107 722B	CFCO 108 799B	-1360-
-1370-	CFCO 109 788B	CFCO 110 7FFB	CFCO 111 E079	CFCO 112 0002	CFCO 113 128E	BMCK 047 2007	BMCK 048 2400	BMCK 049 5E3F	-1370-
				138-					
-1380-	BMCK 050 C4A7	BMCK 051 1212	BMCK 058 5672	BMCK 059 2440	BMCK 060 7672	BMCK 061 2400	BMCK 062 2185	BMCK 063 420F	-1380-
-1390-	BMCK 064 7618	BMCK 065 7318	BMCK 066 5710	BMCK 067 271B	BMCK 068 7718	BMCK 069 5672	BMCK 070 16C5	BMCK 071 1713	-1390-
-13A0-	BMCK 072 6666	BMCK 073 7618	BMCK 074 A7FA	BMCK 075 2404	BMCK 076 2233	BMCK 077 1F2E	BMCK 078 2171	BMCK 079 2807	-13A0-
-13B0-	BMCK 080 7A00	BMCK 081 4632	BMCK 082 261D	BMCK 083 271D	BMCK 084 6662	BMCK 085 17AB	BMCK 086 2185	BMCK 087 873C	-13B0-
-13C0-	BMCK 088 968C	DCLB 106 6224	DCLB 107 6224	DCLB 108 6224	DCLB 109 9414	DCLB 039 1683	DCLB 040 3613	DCLB 041 3773	-13C0-
-13D0-	DCLB 042 5260	DCLB 043 D25A	DCLB 044 1643	DCLB 045 3363	DCLB 046 6224	DCLB 054 0249	DCLB 056 F4C3	DCLB 057 5038	-13D0-
-13E0-	DCLB 058 008B	DCLB 059 FOFA	DCLB 061 5119	DCLB 062 C4C4	DCLB 063 5030	DCLB 071 0049	DCLB 073 F4C5	DCLB 074 0183	-13E0-
-13F0-	DCLB 075 FOC4	DCLB 076 4206	DCLB 077 5038	DCLB 078 008B	DCLB 079 FOC5	DCLB 080 5119	DCLB 081 C4C4	DCLB 082 5049	-13F0-

CONTROL ADDRESS

13--

DATE 11/08/68

ADDRESS LIST

14--

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1400-	DCLB 083 6224	DCLB 084 5038	DCLB 085 0083	DCLB 086 F094	DCLB 087 C51F	DCLB 088 1537	DCLB 089 F031	DCLB 090 E033	-1400-
-1410-	DCLB 091 C035	DCLB 092 D033	DCLB 093 6224	DCLB 095 D639	DCLB 096 3763	DCLB 100 7260	DCLB 101 B082	DCLB 114 2405	-1410-
-1420-	DCLB 115 15F5	DCLB 116 6505	DCLB 117 5038	DCLB 118 4006	DCLB 119 C497	DCLB 121 D63D	DCLB 123 726A	DCLB 124 943E	-1420-
-1430-	DCLB 110 153B	DCLB 111 1518	DCLB 112 153B	DCLB 113 9420	DCLB 103 7212	DCLB 104 B082	DCLB 126 7212	DCLB 128 6226	-1430-
-1440-	DCLB 129 1343	DCLB 131 5230	DCLB 142 D659	DCLB 144 C54A	DCLB 145 1565	DCLB 165 1763	DCLB 166 756A	DCLB 174 6666	-1440-
-1450-	DCLB 175 F25F	DCLB 176 5FF2	DCLB 177 7F60	DCLB 178 9460	DCLB 190 A041	DCLB 191 1D00	DCLB 195 0244	DCLB 179 5F60	-1450-
-1460-	DCLB 180 6664	DCLB 181 42FF	DCLB 183 8721	5007	CMLT 081 5EF9	CMLT 082 53E9	CMLT 083 5239	CMLT 084 5129	-1460-
-1470-	CMLT 085 5019	CMLT 086 5709	CMLT 087 5679	CMLT 088 5869	CMLT 089 5A89	CMLT 090 55A9	CMLT 091 5459	CMLT 092 2407	-1470-
148-									
-1480-	CMLT 093 C50C	CMLT 094 058F	CMLT 095 E08A	CMLT 096 C10C	CMLT 097 C58C	CMLT 098 24F7	CMLT 099 128E	FILE 264 2C15	-1480-
-1490-	FILE 265 2D07	FILE 266 7C62	FILE 267 2A15	FILE 268 3A25	FILE 269 2807	FILE 270 7A52	FILE 271 2A07	FILE 272 2E23	-1490-
-14A0-	FILE 273 498F	FILE 274 128E	FILT 071 50FF	FILT 072 F08D	FILT 073 DB2E	FILT 074 8220	FILT 075 2F43	FILT 076 EB32	-14A0-
-14B0-	FILT 077 2F15	FILT 078 FB36	FILT 079 2F25	FILT 080 0680	FILT 081 2EC3	FILT 082 957A	FILT 083 2F07	FILT 084 C188	-14B0-
-14C0-	FILT 085 E5FE	FILT 086 2D45	FILT 087 4EDF	FILT 088 2E15	FILT 089 4BEF	FILT 090 2E43	FILT 091 4BFF	FILT 092 1D00	-14C0-
-14D0-	FILT 093 4EEF	FILT 094 FB06	FILT 095 B236	FILT 096 C3FD	FILT 097 F059	FILT 098 5E42	FILT 099 6EE6	FILT 100 6EE6	-14D0-
-14E0-	FILT 101 5CF8	FILT 102 4DCF	FILT 103 2C63	FILT 104 4FCF	FILT 105 5C52	FILT 106 2C07	FILT 107 F8EC	FILT 108 7C52	-14E0-
-14F0-	FILT 109 0E04	FILT 110 3D00	FILT 111 2C15	FILT 112 4BCF	FILT 113 6EE4	FILT 114 C3FB	FILT 115 0222	FILT 116 CA86	-14F0-

CONTROL ADDRESS

14--

ADDRESS LIST

DATE 11/08/68

15--

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1500-	FILT 117 DF87	FILT 118 48FF	FILT 119 A37E	FILT 120 5ADF	FILT 121 C48C	FILT 122 5C62	FILT 123 CA9F	FILT 124 C58C	-1500-
-1510-	FILT 125 58CF	FILT 126 6CD1	FILT 127 F098	FILT 128 8EOC	FILT 129 2D07	FILT 130 2C13	FILT 131 956A	FILT 132 5E52	-1510-
-1520-	FILT 133 EB31	FILT 134 CD32	FILT 135 2D1B	FILT 136 7C62	FILT 137 5DF9	FILT 138 0DAB	FILT 139 F0B5	FILT 140 8DD4	-1520-
-1530-	FILT 141 A3D8	FILT 142 869C	FILT 143 2D25	FILT 144 2C07	FILT 145 2F07	FILT 146 956A	FILT 147 2D07	FILT 148 2C45	-1530-
-1540-	FILT 149 DEEA	FILT 150 2E85	FILT 151 EED9	FILT 152 FA9B	FILT 153 5ACF	FILT 154 C4DC	FILT 155 7C62	FILT 156 2EC3	-1540-
-1550-	FILT 157 D8F5	FILT 158 C8F7	FILT 159 2D07	FILT 160 2E15	FILT 161 2C15	FILT 162 956C	FILT 163 5CDB	FILT 164 1CF5	-1550-
-1560-	FILT 165 ED64	FILT 166 3D83	FILT 167 FD68	FILT 168 3D23	FILT 169 1D35	FILT 170 2E07	FILT 171 7E62	FILT 172 2EE3	-1560-
-1570-	FILT 173 3C25	FILT 174 D8F6	FILT 175 3E13	FILT 176 7C52	FILT 177 2F07	FILT 178 FDFE	FILT 179 3E45	FILT 180 C982	-1570-
158-									
-1580-	FILT 181 3F85	FILT 182 1D04	FILT 183 C88D	FILT 184 F98F	FILT 185 4386	FILT 186 C48F	FILT 187 3F45	FILT 188 2A07	-1580-
-1590-	FILT 189 0F83	FILT 190 C496	FILT 191 EA20	FILT 192 2B43	FILT 193 3B15	FILT 194 5C42	FILT 195 7EA8	FILT 196 7CA0	-1590-
-15A0-	FILT 197 8E40	CFMU 062 C627	CFMU 063 0080	CFMU 064 27C3	CFMU 065 B266	CFMU 024 5CA2	CFMU 025 C93D	CFMU 026 3D83	-15A0-
-1580-	CFMU 027 7CA2	CFMU 028 5CE2	CFMU 029 52C8	CFMU 030 5EC0	CFMU 031 2C13	CFMU 032 8356	CFMU 033 54B2	CFMU 034 5AF9	-1580-
-15C0-	CFMU 035 5B89	CFMU 036 5699	CFMU 037 5359	CFMU 038 5039	CFMU 039 5109	CFMU 040 5219	CFMU 041 5729	CFMU 042 46C6	-15C0-
-15D0-	CFMU 043 5CC2	CFMU 044 0FF3	CFMU 045 C4DE	CFMU 046 A1E4	CFMU 047 26FF	CFMU 048 F4DF	CFMU 049 3713	CFMU 050 5AE2	-15D0-
-15E0-	CFMU 051 8BDA	CFMU 052 D5E7	CFMU 053 B260	CFMU 054 56E9	CFMU 055 1E85	CFMU 056 C76E	CFMU 057 3E85	CFMU 058 2080	-15E0-
-15F0-	CFMU 059 F322	CFMU 060 27D3	CFMU 061 B25E	CFMU 009 5CB2	CFMU 010 FC2A	CFMU 011 44C6	CFMU 012 1415	CFMU 013 5CA2	-15F0-

CONTROL ADDRESS

15--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1600-	CFMU 014 5079	CFMU 015 5F09	CFMU 016 57F9	CFMU 017 5189	CFMU 018 5299	CFMU 019 5329	CFMU 020 5E39	CFMU 021 2107	-1600-
-1610-	CFMU 022 2507	CFMU 023 95CE	DCLR 108 3415	DCLR 109 15C5	DCLR 110 1533	DCLR 112 81CA	ALDP 022 26A3	ALDP 023 2783	-1610-
-1620-	ALDP 024 873C	ALDP 025 8744	ALDP 026 3643	ALDP 027 043D	ALDP 028 EOC3	ALDP 029 873C	ALDP 030 2483	ALDP 031 2577	-1620-
-1630-	ALDP 032 534A	ALDP 033 874E	ALDP 034 D131	ALDP 035 D01D	ALDP 036 7543	ALDP 037 F530	ALDP 038 24F3	ALDP 039 F330	-1630-
-1640-	ALDP 040 9638	ALDP 041 044B	ALDP 042 FOCF	ALDP 043 041B	ALDP 044 FOAA	ALDP 045 3625	ALDP 046 3613	ALDP 047 8746	-1640-
-1650-	ALDP 048 3645	ALDP 071 042D	ALDP 072 E0DC	ALDP 073 3715	ALDP 074 047B	ALDP 075 9668	ALDP 076 043F	ALDP 077 EOAA	-1650-
-1660-	ALDP 078 FOE6	ALDP 079 E66C	ALDP 080 C9ED	ALDP 081 041B	ALDP 082 FOAA	ALDP 083 3615	ALDP 084 6662	ALDP 085 874C	-1660-
-1670-	ALDP 086 8746	ALDP 087 5403	ALDP 088 8746	ALDP 089 440D	ALDP 090 8746	ALDP 091 5413	ALDP 092 8746	ALDP 093 441D	-1670-
168-									
-1680-	ALDP 094 873C	ALDP 095 FOOA	ALDP 096 874C	ALDP 097 874C	ALDP 098 17AD	ALDP 099 2243	ALDP 100 2515	ALDP 104 F737	-1680-
-1690-	ALDP 108 F61B	ALDP 112 00C9	ALDP 114 F49A	ALDP 115 3713	ALDP 116 962C	ALDP 117 3685	ALDP 118 E63A	ALDP 119 8746	-1690-
-16A0-	ALDP 120 54F3	ALDP 121 8746	ALDP 122 44FD	ALDP 123 172D	ALDP 124 839C	ALDP 125 E70F	ALDP 126 874C	ALDP 127 25FF	-16A0-
-16B0-	ALDP 128 C48E	ALDP 129 873C	ALDP 130 968C	ALDP 148 0049	ALDP 150 9694	ALDP 152 839C	ALDP 153 1645	ALDP 154 C744	-16B0-
-16C0-	ALDP 155 1785	ALDP 156 5FE9	ALDP 157 5E35	ALDP 158 23FD	ALDP 159 0361	ALDP 160 F4D0	ALDP 161 237B	ALDP 162 23CD	-16C0-
-16D0-	ALDP 163 874E	ALDP 164 D75B	ALDP 165 5E3D	ALDP 166 176D	ALDP 167 96C6	ALDP 168 174D	ALDP 169 E743	ALDP 170 22FF	-16D0-
-16E0-	ALDP 171 C4AC	ALDP 172 3743	ALDP 173 874E	ALDP 174 1743	ALDP 175 C36F	ALDP 176 873C	ALDP 177 8752	ALDP 178 2243	-16E0-
-16F0-	ALDP 179 8744	ALDP 180 96AE	DYPE 191 2791	DYPE 192 5860	DYPE 193 58BB	DYPE 194 7B60	DYPE 195 6664	DYPE 196 2404	-16F0-

CONTROL ADDRESS

16--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1700-	DYPE 197 A078	DYPE 198 0F04	DYPE 199 D111	DYPE 200 C413	DYPE 201 EE8D	DYPE 202 B362	DYPE 203 8954	DYPE 208 8938	-1700-
-1710-	DYPE 204 895E	DYPE 205 3623	DYPE 206 CE0F	DYPE 207 8924	5007	FILX 248 2C07	FILX 249 2D35	FILX 250 3D43	-1710-
-1720-	FILX 251 7EC0	FILX 252 85EE	FILX 238 978E	FILX 211 F92E	FILX 212 2C07	FILX 213 2D07	FILX 214 7C62	FILX 215 7CE8	-1720-
-1730-	FILX 216 5C62	FILX 217 5EDF	FILX 218 1DA3	FILX 219 DD3A	FILX 220 3D83	FILX 221 7CE0	FILX 222 2C07	FILX 223 2D07	-1730-
-1740-	FILX 224 3F15	FILX 225 7CEA	FILX 226 1F1D	FILX 227 FBF	FILX 228 5DE8	FILX 229 D5CF	FILX 230 7DB8	FILX 231 5886	-1740-
-1750-	FILX 232 C4FD	FILX 233 CB46	FILX 234 1FEB	FILX 235 FF44	FILX 236 FBA5	FILX 237 B22A	FILX 199 1C25	FILX 200 7C52	-1750-
-1760-	FILX 201 5E32	FILX 202 1E45	FILX 203 7E32	FILX 204 2E07	FILX 205 2F25	FILX 206 3F43	FILX 207 FBA6	FILX 208 5DB8	-1760-
-1770-	FILX 209 7DE8	FILX 210 974E	FILX 195 3E00	FILX 196 5FFF	FILX 197 E0DD	FILX 198 86AC	FILX 239 CD98	FILX 240 CB06	-1770-
178-									
-1780-	FILX 241 FF06	FILX 242 F88F	FILX 243 B22A	FILX 244 2F45	FILX 245 FBC9	FILX 246 F985	FILX 247 B208	FILX 261 2A07	-1780-
-1790-	FILX 262 2B25	FILX 263 3B43	FILX 264 5CA8	FILX 265 4CC6	FILX 266 C4A4	FILX 267 5CA0	FILX 268 3B15	FILX 269 5EAA	-1790-
-17A0-	FILX 270 6EC5	FILX 271 C4AF	FILX 272 2C13	FILX 273 2F45	FILX 274 CDC7	FILX 275 F9C2	FILX 276 97CC	FILX 280 5DE9	-17A0-
-17B0-	FILX 281 3B13	FILX 282 6BE3	FILX 283 F4BA	FILX 284 0EFF	FILX 285 C4A4	FILX 286 2CA3	FILX 287 7CF3	FILX 288 F4A4	-17B0-
-17C0-	FILX 289 8DCE	FILX 277 4886	FILX 278 C4AD	FILX 279 97CE	FILX 253 2C13	FILX 254 F9CE	FILX 255 2F07	FILX 256 3CA5	-17C0-
-17D0-	FILX 257 2D07	FILX 258 7C52	FILX 259 2EE3	FILX 260 8E3A	CDVD 088 2002	CDVD 089 711B	CDVD 090 7G0B	CDVD 091 733B	-17D0-
-17E0-	CDVD 092 722B	CDVD 093 799B	CDVD 094 788B	CDVD 095 755B	CDVD 096 744B	CDVD 097 128E	CDVD 100 2002	CDVD 101 F5E3	-17E0-
-17F0-	CDVD 102 0008	CDVD 103 3002	CDVD 104 711B	CDVD 105 700B	CDVD 106 733B	CDVD 107 722B	CDVD 108 128E	5007	-17F0-

CONTROL ADDRESS

17--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1800-	ECOL 083 2F08	ECOL 084 980A	ECOL 085 2F0C	ECOL 086 980A	ECOL 087 2F0A	ECOL 088 221B	ECOL 089 F491	ECOL 090 3C13	-1800-
-1810-	ECOL 091 911A	ERDR 053 54E9	ERDR 054 C4A0	ERDR 055 5EB2	ERDR 056 5D69	ERDR 057 A044	ERDR 058 3C85	ERDR 059 930E	-1810-
-1820-	ERDR 060 C3AE	ERDR 061 F72C	ERDR 062 2406	ERDR 063 CEAC	ERDR 064 3D23	ERDR 065 DDAF	ERDR 066 9EE0	ERDR 067 8DC0	-1820-
-1830-	ERDR 068 A044	ERDR 069 CEBD	ERDR 070 9258	ERDR 071 2C75	ERDR 072 4452	ERDR 073 9D54	ERDR 074 5460	ERDR 075 F747	-1830-
-1840-	ERDR 076 2D80	ERDR 077 0F30	ERDR 078 984A	ERDR 079 2D04	ERDR 080 0B30	ERDR 081 2FA3	ERDR 082 0E3B	ERDR 083 C4DC	-1840-
-1850-	ERDR 084 F758	ERDR 085 DDDA	ERDR 086 B16A	ERDR 087 861C	ERDR 088 CFD5	ERDR 089 98B6	ERDR 090 0E4B	ERDR 091 F76F	-1850-
-1860-	ERDR 092 FOE9	ERDR 093 2D07	ERDR 094 CFDA	ERDR 095 92D4	ERDR 096 CFF3	ERDR 097 3D45	ERDR 098 9874	ERDR 099 FOFA	-1860-
-1870-	ERDR 100 DDEA	ERDR 101 1D45	ERDR 102 B16A	ERDR 103 3623	ERDR 104 A4E8	ERDR 105 2D07	ERDR 106 DDDA	ERDR 107 1B00	-1870-
188-									
-1880-	ERDR 108 6CF7	ERDR 109 EA34	ERDR 110 FA26	ERDR 111 EE34	ERDR 112 E835	ERDR 113 76F2	ERDR 114 CE1A	ERDR 115 DE18	-1880-
-1890-	ERDR 116 FA35	ERDR 117 OD04	ERDR 118 A078	ERDR 119 910A	ERDR 120 3C45	ERDR 121 DE1E	ERDR 122 3C25	ERDR 123 1C83	-1890-
-18A0-	ERDR 124 3C23	ERDR 125 8964	ERDR 126 9C03	ERDR 127 DA35	ERDR 128 EDAF	ERDR 129 2B40	ERDR 130 9304	ERDR 131 3D00	-18A0-
-18B0-	ERDR 133 2E13	ERDR 137 98B8	ERDR 138 2D85	ERDR 139 2E23	ERDR 140 8964	ERDR 141 B16A	ERDR 142 2A07	ERDR 143 C3C2	-18B0-
-18C0-	ERDR 144 9EE0	ERDR 145 9D3C	ERDR 146 CC5D	ERDR 148 F753	ERDR 149 2445	ERDR 150 2513	ERDR 151 40A6	ERDR 152 31BB	-18C0-
-18D0-	ERDR 153 905A	ERDR 154 EE56	ERDR 158 36A9	ERDR 159 2855	ERDR 160 2915	ERDR 161 3983	ERDR 162 5E49	ERDR 164 F565	-18D0-
-18E0-	ERDR 168 2A35	ERDR 169 9BE6	ERDR 170 261B	ERDR 171 F4EA	ERDR 172 98B6	ERDR 173 5884	ERDR 174 E471	ERDR 175 5884	-18E0-
-18F0-	ERDR 176 21FF	ERDR 177 C4E4	ERDR 178 98B8	CSTS 093 55B9	CSTS 094 2125	CSTS 095 2F07	CSTS 096 217B	CSTS 097 9C76	-18F0-

CONTROL ADDRESS

18--

ADDRESS LIST		DATE 11/08/68							19--
CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1900-	ERDR 180 2808	ERDR 181 9050	ERDR 182 280C	ERDR 183 9050	ERDR 184 280A	ERDR 185 9050	ECOL 041 75C0	ECOL 042 2DAB	-1900-
-1910-	ECOL 043 78C0	ECOL 044 2DAB	ECOL 045 7FC0	ECOL 046 1C1B	ECOL 047 F871	ECOL 048 5959	ECOL 049 5AB9	ECOL 050 5EF9	-1910-
-1920-	ECOL 051 2DAB	ECOL 052 990C	ECOL 022 5738	ECOL 023 E72A	ECOL 024 6541	ECOL 025 F72E	ECOL 026 6B41	ECOL 027 C332	-1920-
-1930-	ECOL 028 6F41	ECOL 029 D336	ECOL 030 6941	ECOL 031 E33A	ECOL 032 6A41	ECOL 033 F33E	ECOL 034 6E41	ECOL 035 5224	-1930-
-1940-	ECOL 036 282B	ECOL 037 311B	ECOL 038 C48D	ECOL 039 5879	ECOL 040 B345	ECOL 003 2485	ECOL 004 2AF7	ECOL 005 5AB9	-1940-
-1950-	ECOL 006 4EA6	ECOL 007 5B99	ECOL 008 5859	ECOL 009 01E3	ECOL 010 F0E1	ECOL 011 211B	ECOL 012 D861	ECOL 013 311B	-1950-
-1960-	ECOL 014 E824	ECOL 015 261B	ECOL 016 F4A4	ECOL 017 2DA1	ECOL 018 5ECO	ECOL 019 3E13	ECOL 020 7ECO	ECOL 021 997C	-1960-
-1970-	ECOL 053 2DCD	ECOL 054 2DFB	ECOL 055 5119	ECOL 056 C4FD	ECOL 057 08AD	ECOL 058 EOCA	ECOL 059 1C2B	ECOL 060 E814	-1970-
198-									
-1980-	ECOL 061 5629	ECOL 062 71A2	ECOL 063 5C99	ECOL 064 A062	ECOL 065 A044	ECOL 066 2E07	ECOL 067 C538	ECOL 068 F837	-1980-
-1990-	ECOL 069 52D9	ECOL 070 93CA	ECOL 092 5089	ECOL 093 51A2	ECOL 094 52E2	ECOL 095 D81E	ECOL 096 5224	ECOL 097 01E3	-1990-
-19A0-	ECOL 098 FOAB	ECOL 099 21FF	ECOL 100 D829	ECOL 101 212B	ECOL 102 1C4B	ECOL 103 58D5	ECOL 104 2DCB	ECOL 105 2D3D	-19A0-
-19B0-	ECOL 106 5889	ECOL 107 FOCO	ECOL 108 994A	ECOL 071 1525	ECOL 072 3485	ECOL 080 5FA0	ECOL 081 7CA8	ECOL 082 9F01	-19B0-
-19C0-	ECOL 109 2D3D	ECOL 110 2D2B	ECOL 111 5FC0	ECOL 112 3D9B	ECOL 113 5BC0	ECOL 114 3D9B	ECOL 115 55C0	ECOL 116 1C1B	-19C0-
-19D0-	ECOL 117 F854	ECOL 118 9946	ECOL 119 5FE9	ECOL 120 5BA9	ECOL 121 5599	ECOL 122 3D9B	ECOL 123 99C4	ERCX 056 3B23	-19D0-
-19E0-	ERCX 057 9BDE	ERCX 058 9BDA	ERCX 059 99DE	ERCX 060 99EE	ERCX 061 9BDC	ERCX 062 99EE	ERCX 063 99EE	ERCX 067 F775	-19E0-
-19F0-	ERCX 068 2F80	ERCX 069 9BDE	ERCX 070 2B80	ERCX 072 9BDE	CDVD 002 E1FD	CDVD 003 3485	CDVD 004 837C	CDVD 005 A21E	-19F0-

CONTROL ADDRESS

19--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1A00-	DPTC 029 C82E	DPTC 030 9835	DPTC 031 8612	DPTC 032 8612	DPTC 033 8612	DPTC 034 8612	DPTC 035 8612	DPTC 036 9042	-1A00-
-1A10-	DPTC 037 C80C	DPTC 038 9835	DPTC 039 9835	DPTC 040 9835	DPTC 041 9835	DPTC 042 9835	DPTC 043 C81A	DPTC 044 0C3B	-1A10-
-1A20-	DPTC 045 FOA4	DPTC 047 AF7A	DPTC 051 0CBB	DPTC 052 F08C	DPTC 053 FC2D	DPTC 055 860C	DPTC 060 AF78	DPTC 065 9855	-1A20-
-1A30-	DCLE 032 FDB8	DCLE 033 929E	DCLE 030 EDB0	DCLE 031 8172	DCLE 034 2840	DCLE 035 2645	DCLE 036 3613	DCLE 037 92B8	-1A30-
-1A40-	ECOL 124 C862	ECOL 125 990C	ECOL 126 2445	ECOL 127 9A5E	ECOL 128 2425	ECOL 129 9A5E	ECOL 130 2415	ECOL 131 9A5E	-1A40-
-1A50-	ECOL 132 2483	ECOL 133 9A5E	ECOL 134 2443	ECOL 135 9A5E	ECOL 136 2423	ECOL 137 9A5E	ECOL 138 2413	ECOL 139 C864	-1A50-
-1A60-	ECOL 140 9956	ECOL 141 9830	ECOL 142 9B66	CICY 079 2F07	CICY 080 128E	CICY 070 5098	CICY 071 50BB	CICY 072 C4E7	-1A60-
-1A70-	CICY 073 5EA8	CICY 074 56A0	CICY 075 617B	CICY 076 606F	CICY 077 6FAD	CICY 078 128E	CBIN 038 ADA8	CBIN 039 AC1A	-1A70-
1A8-									
-1A80-	EPXF 041 2DAB	EPXF 042 2DAB	EPXF 043 2DAB	EPXF 044 2DAB	EPXF 045 2DAB	EPXF 046 2DAB	EPXF 047 9B08	EPXF 048 9814	-1A80-
-1A90-	DYPE 180 5E29	DYPE 181 2791	DYPE 182 7C68	DYPE 183 7468	DYPE 184 7268	DYPE 185 7068	DYPE 186 2771	DYPE 187 128E	-1A90-
-1AA0-	DPTC 083 8612	DPTC 084 8CDC	DPTC 085 8612	DPTC 086 8D54	DPTC 087 8612	DPTC 088 8612	DPTC 089 8612	DPTC 090 8612	-1AA0-
-1AB0-	CLST 103 AD26	CLST 104 5218	CLST 105 72A8	CLST 106 5218	CLST 107 72A0	CLST 108 AD30	CLST 109 9AB2	5007	-1AB0-
-1AC0-	DPTC 091 8612	DPTC 092 8CDC	DPTC 093 8612	DPTC 094 861C	DPTC 095 8F66	DPTC 096 8612	DPTC 097 8612	DPTC 098 8612	-1AC0-
-1AD0-	CFCO 078 7EA8	CFCO 079 78A8	CFCO 080 F45B	CFCO 081 72A8	CFCO 082 70A0	CFCO 083 1B73	CFCO 084 128E	5007	-1AD0-
-1AE0-	CSFT 046 9C34	CSFT 047 2D2B	CSFT 048 9AE8	CSFT 049 1D23	CSFT 050 A460	CSFT 051 B04C	CSFT 052 9C34	COMD 213 3002	-1AE0-
-1AF0-	COMD 214 56F2	COMD 215 4206	COMD 216 5D30	COMD 217 5F6A	COMD 218 7DFF	COMD 219 7D3A	COMD 220 C375	COMD 221 128E	-1AF0-

CONTROL ADDRESS

1A--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-1E00-	CFMD 075 8312	DCLC 149 93DA	DCLC 112 A254	DCLC 085 E605	DCLC 086 FD8F	DCLC 087 262B	DCLC 088 FD86	DCLC 089 5F4F	-1E00-
-1E10-	DCLC 090 FC85	DCLC 091 E123	DCLC 092 F123	DCLC 093 2888	DCLC 094 E605	DCLC 095 C9A0	DCLC 096 262B	DCLC 097 C999	-1E10-
-1E20-	DCLC 098 884C	DCLC 099 C48E	DCLC 100 F233	DCLC 101 2D04	DCLC 102 3685	DCLC 103 7632	DCLC 104 1685	DCLC 105 D134	-1E20-
-1E30-	DCLC 106 2D02	DCLC 107 1543	DCLC 108 2B48	DCLC 109 1635	DCLC 110 C3BD	DCLC 111 A7C0	DCLC 113 8AD0	DCLC 114 C050	-1E30-
-1E40-	DCLC 115 F451	DCLC 116 2D04	DCLC 117 D14A	DCLC 118 1543	DCLC 119 2D02	DCLC 120 2285	DCLC 121 2307	DCLC 122 7232	-1E40-
-1E50-	DCLC 123 2307	DCLC 124 C561	DCLC 125 D562	DCLC 126 04C7	DCLC 127 C4E2	DCLC 128 C463	DCLC 129 D065	DCLC 130 C07B	-1E50-
-1E60-	DCLC 131 15C5	DCLC 132 A782	DCLC 138 2D10	DCLC 139 2888	DCLC 140 E605	DCLC 141 C9F0	DCLC 142 262B	DCLC 143 C9E9	-1E60-
-1E70-	DCLC 144 2645	DCLC 145 3613	DCLC 146 5212	DCLC 147 D402	DCLC 148 93D6	DCLC 133 2D10	DCLC 134 2848	DCLC 135 2405	-1E70-
1E8-									
-1E80-	DCLC 136 1573	DCLC 137 9E38	ERRQ 053 8924	ERRQ 064 2406	ERRQ 065 0B30	ERRQ 066 8DC4	ERRQ 067 A044	ERRQ 068 2D04	-1E80-
-1E90-	ERRQ 069 989E	ERRQ 012 C46D	ERRQ 013 CFB9	ERRQ 014 8954	ERRQ 054 E263	ERRQ 055 C3C9	ERRQ 057 C424	ERRQ 058 D129	-1E90-
-1EA0-	ERRQ 059 CA25	ERRQ 060 DA29	ERRQ 061 3423	ERRQ 062 15C5	ERRQ 063 9D52	ERRQ 004 A06E	ERRQ 005 3623	ERRQ 006 CEB3	-1EA0-
-1EB0-	ERRQ 007 9836	ERRQ 008 F712	ERRQ 009 C43D	ERRQ 010 DD96	ERRQ 011 B362	ERRQ 029 A06A	ERRQ 030 2466	ERRQ 031 E607	-1EB0-
-1EC0-	ERRQ 032 2406	ERRQ 033 D998	ERRQ 034 0B10	ERRQ 035 3485	ERRQ 036 B11A	ERRQ 037 D54E	ERRQ 038 9D44	ERRQ 039 3E43	-1EC0-
-1ED0-	ERRQ 040 E263	ERRQ 041 F728	ERRQ 042 C3A8	ERRQ 043 D461	ERRQ 044 EA61	ERRQ 045 4452	ERRQ 046 A568	ERRQ 047 8776	-1ED0-
-1EE0-	ERRQ 048 5007	ERRQ 049 2A07	ERRQ 050 B11A	ERRQ 051 D104	ERRQ 052 895E	ERRQ 016 A062	ERRQ 017 DB98	ERRQ 018 0F10	-1EE0-
-1EF0-	ERRQ 019 C446	ERRQ 020 FFC8	ERRQ 021 8DC4	ERRQ 022 2D15	ERRQ 023 5CCD	ERRQ 024 8964	ERRQ 025 7CA0	ERRQ 026 1545	-1EF0-

CONTROL ADDRESS

1E--

ADDRESS LIST

DATE 11/08/68

20--

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2000-	BDIA 091 2810	BDIA 092 25F5	BDIA 096 E504	BDIA 097 2810	BDIA 101 F508	BDIA 102 2810	BDIA 103 A040	CCOM 110 2607	-2000-
-2010-	CCOM 114 577B	CCOM 115 3763	CCOM 116 5B60	CCOM 117 5B6B	CCOM 118 2707	CCOM 120 FOA5	CCOM 121 4E52	CCOM 123 4252	-2010-
-2020-	CCOM 125 4626	CCOM 126 4026	CCOM 127 2B45	CCOM 129 42AF	CCOM 131 76B8	CCOM 132 72B8	CCOM 133 6AA4	CCOM 134 70B0	-2020-
-2030-	CCOM 135 2B47	CCOM 137 42AF	CCOM 139 7E80	CCOM 141 5282	CCOM 142 423F	CCOM 144 C5BE	CCOM 145 128E	CCOM 146 ADF4	-2030-
-2040-	BDIA 106 2810	BDIA 111 C55F	ERRQ 092 2A15	ERRQ 093 F750	ERRQ 094 2406	ERRQ 095 2BF5	ERRQ 096 5CAA	ERRQ 097 128E	-2040-
-2050-	ERRQ 098 2402	ERRQ 099 2BB1	ERRQ 100 C4CC	CICY 064 6884	CICY 065 6884	CICY 066 2B13	CICY 067 9C70	BDIA 114 2810	-2050-
-2060-	BDIA 119 D57F	ERRQ 087 4642	ERRQ 088 3725	ERRQ 089 2402	ERRQ 090 C4F2	ERRQ 075 4642	ERRQ 076 3715	ERRQ 077 F766	-2060-
-2070-	ERRQ 078 2406	ERRQ 080 3743	ERRQ 081 5060	ERRQ 082 1743	ERRQ 083 5468	ERRQ 084 5E6A	ERRQ 085 128E	BDIA 125 2800	-2070-
208-									
-2080-	BDIA 131 3E09	BDIA 132 C483	BDIA 133 F085	BDIA 134 E087	BDIA 135 F489	BDIA 136 6EE5	BDIA 137 A0DA	BDIA 138 2E0D	-2080-
-2090-	BDIA 142 A0DA	BDIA 143 7EE1	BDIA 147 F494	BDIA 148 7EE1	BDIA 149 6EE3	BDIA 150 2E5F	BDIA 151 6EE3	BDIA 152 6EE9	-2090-
-20A0-	BDIA 153 0E5F	BDIA 154 C4A2	BDIA 155 2E87	BDIA 156 3E27	BDIA 157 2EEF	BDIA 158 2E77	BDIA 159 1E87	BDIA 160 1EB7	-20A0-
-20B0-	BDIA 161 2EF7	BDIA 162 0E11	BDIA 163 F0B4	BDIA 164 E0B6	BDIA 165 1EFB	BDIA 166 A0E2	BDIA 167 1EF3	BDIA 171 A0E2	-20B0-
-20C0-	BDIA 172 3E15	BDIA 176 A0E2	BDIA 177 3E0D	BDIA 181 2E1D	BDIA 182 2FF5	BDIA 183 A0DA	BDIA 184 OFFD	BDIA 188 C4CE	-20C0-
-20D0-	BDIA 189 1FF5	BDIA 190 C4D2	BDIA 191 2EF5	BDIA 192 A0E2	BDIA 193 A0E8	BDIA 195 0EFF	BDIA 196 C4E1	BDIA 197 2810	-20D0-
-20E0-	BDIA 198 128E	BDIA 201 0EFD	BDIA 202 C4DE	BDIA 203 128E	BDIA 207 24F6	BDIA 208 578F	BDIA 209 1887	BDIA 210 087F	-20E0-
-20F0-	BDIA 211 E0F0	BDIA 212 F0F2	BDIA 213 2400	BDIA 214 574F	BDIA 215 0487	BDIA 216 E0FA	BDIA 217 F0FC	BDIA 221 2EF7	-20F0-

CONTROL ADDRESS

20--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2100-	BDIA 222 2FF7	BDIA 223 6E02	BDIA 224 2E05	BDIA 225 2F05	BDIA 226 4E02	BDIA 227 0FFF	BDIA 228 E08C	BDIA 229 F08E	-2100-
-2110-	BDIA 230 1EFF	BDIA 231 C492	BDIA 232 2E33	BDIA 233 2F87	BDIA 234 6FE8	BDIA 235 6FEA	BDIA 236 44E8	BDIA 237 2F13	-2110-
-2120-	BDIA 238 048F	BDIA 239 E0A2	BDIA 240 F0A4	BDIA 241 257F	BDIA 242 C4A8	BDIA 243 2E05	BDIA 244 55EA	BDIA 245 C4AE	-2120-
-2130-	BDIA 246 2E45	BDIA 247 3FE9	BDIA 248 25F7	BDIA 249 75E0	BDIA 251 2505	BDIA 252 55EA	BDIA 254 05FF	BDIA 255 C48E	-2130-
-2140-	BDIA 256 FB34	BDIA 257 44E6	BDIA 258 044D	BDIA 259 C4C6	BDIA 260 75F1	BDIA 261 C4CA	BDIA 262 0060	BDIA 263 88EC	-2140-
-2150-	CFLS 014 1E85	CFLS 021 1E8D	CFLS 022 2C25	CFLS 023 CE5A	CFLS 024 1C7D	CFLS 025 8BDA	CFLS 026 D5E1	CFLS 027 2C07	-2150-
-2160-	CFLS 028 B266	DCHN 092 2400	DCHN 093 2A07	DCHN 094 C3EA	DCHN 095 875A	DCHN 096 A8DE	BSYS 002 3210	BSYS 004 2460	-2160-
-2170-	BSYS 005 2607	BSYS 018 24F6	BSYS 019 4052	BSYS 020 4206	BSYS 021 4406	BSYS 022 4606	BSYS 026 2D08	BSYS 028 2490	-2170-
218-									
-2180-	BSYS 029 AFEA	BSYS 030 8224	BSYS 031 49CF	BSYS 032 E581	BSYS 033 2E08	BSYS 037 2482	BSYS 039 4A52	BSYS 040 4E42	-2180-
-2190-	BSYS 042 5EE6	BSYS 043 57E8	BSYS 044 497F	BSYS 045 2400	BSYS 046 6EE4	BSYS 047 56E0	BSYS 048 2607	BSYS 049 577D	-2190-
-21A0-	BSYS 050 17B3	BSYS 051 76EA	BSYS 053 7AE0	BSYS 054 2F1D	BSYS 055 F4A4	BSYS 056 2F91	BSYS 057 7AEA	BSYS 059 2E15	-21A0-
-21B0-	BSYS 060 7AEA	BSYS 061 2FF5	BSYS 062 7AE0	BSYS 064 7A32	BSYS 066 7A52	BSYS 067 7A62	BSYS 118 40A6	BSYS 119 3212	-21B0-
-21C0-	BSYS 120 1210	BSYS 121 20A0	BSYS 122 C3C8	BSYS 123 A5DA	BSYS 124 E8D2	BSYS 125 062C	BSYS 126 1212	BSYS 127 4052	-21C0-
-21D0-	BSYS 128 9C9A	BSYS 129 162C	BSYS 130 1212	BSYS 131 5C82	BSYS 132 44CF	BSYS 133 42DF	BSYS 134 4852	BSYS 135 5CC2	-21D0-
-21E0-	BSYS 136 2F07	BSYS 137 9CB2	CFCO 034 5FF3	CFCO 035 48F5	CFCO 036 5883	CFCO 037 4985	CFCO 038 5993	CFCO 039 F474	-21E0-
-21F0-	CFCO 040 4595	CFCO 041 128E	CFCO 042 4295	CFCO 043 5223	CFCO 044 4325	CFCO 045 5333	CFCO 046 4035	CFCO 047 5003	-21F0-

CONTROL ADDRESS

21--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2300-	DCLT 050 C407	DCLT 051 E98A	DCLT 052 A5FC	DCLT 054 D985	DCLT 055 C985	DCLT 056 5A02	DCLT 057 2A15	DCLT 058 7A22	-2300-
-2310-	DCLT 059 2A13	DCLT 060 C446	DCLT 061 2A07	DCLT 062 2413	DCLT 063 24F6	DCLT 064 221E	DCLT 065 F056	DCLT 066 1413	-2310-
-2320-	DCLT 067 AF18	DCLH 015 8AF6	DCLH 016 2C25	DCLH 017 24F6	DCLH 018 2210	DCLH 019 F030	DCLH 020 0210	DCLH 021 8362	-2320-
-2330-	DCLH 022 1D00	DCLH 023 2884	DCLH 024 0210	DCLH 025 E53F	DCLH 026 4006	DCLH 027 C48F	DCLH 028 3445	DCLH 029 8080	-2330-
-2340-	DCLH 004 C625	DCLH 005 2043	DCLH 006 7022	DCLH 007 24F6	DCLH 008 2880	DCLH 009 5622	DCLH 010 5659	DCLH 011 2645	-2340-
-2350-	DCLH 012 2407	DCLH 013 D122	DCLH 014 8AFA	DCLT 068 1583	DCLT 069 FCDC	DCLT 070 3483	DCLT 071 0D40	DCLT 072 2470	-2350-
-2360-	DCLT 073 74A2	DCLT 074 70F2	DCLT 075 2400	DCLT 076 5BA2	DCLT 077 3885	DCLT 078 7AE2	DCLT 079 AF2C	CMLT 168 3002	-2360-
-2370-	CMLT 169 3D25	CMLT 170 9CD2	CMLT 171 1D25	CMLT 172 3002	CMLT 173 9CD4	CMLT 174 8344	5007	FILX 295 1D00	-2370-
238-									
-2380-	FILX 296 E964	FILX 297 5C52	FILX 298 F590	FILX 299 2D23	FILX 300 3D15	FILX 301 1D00	FILX 302 48DF	FILX 303 0222	-2380-
-2390-	FILX 310 FFDF	FILX 311 CBD7	FILX 312 DBBA	FILX 313 C959	FILX 314 FFE2	FILX 315 2D07	FILX 316 48DF	FILX 317 2D23	-2390-
-23A0-	FILX 318 4EDF	FILX 319 2D18	FILX 320 FOA2	FILX 321 48DF	FILX 322 5E32	FILX 323 1D00	FILX 324 EE0E	FILX 325 2E85	-23A0-
-2380-	FILX 326 7E32	FILX 327 C387	FILX 328 B362	FILX 329 94D8	FILX 337 DB96	FILX 338 5E32	FILX 339 1E45	FILX 340 FFCE	-2380-
-23C0-	FILX 341 F9CA	FILX 342 CDCB	FILX 343 DDCA	FILX 344 7E32	FILX 345 B238	FILX 350 7E32	FILX 351 B22A	FILX 346 F5C1	-23C0-
-23D0-	FILX 347 CDCD	FILX 348 DDC7	FILX 349 B206	FILX 333 D962	FILX 334 2E25	FILX 335 2F43	FILX 336 868A	FILX 330 C888	-23D0-
-23E0-	FILX 331 E959	FILX 332 9760	FILX 304 CDE9	FILX 305 F9E8	FILX 306 8E24	FILX 307 DD8F	FILX 308 2E83	FILX 309 8E38	-23E0-
-23F0-	5007	DCHN 031 56D2	DCHN 032 1773	DCHN 037 57B5	DCHN 038 1883	DCHN 039 3885	DCHN 040 58A0	DCHN 043 5468	-23F0-

CONTROL ADDRESS

23--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2400-	DCHN 044 C251	DCHN 045 B320	DCHN 046 4052	DCHN 047 4206	DCHN 048 D146	DCHN 049 C110	DCHN 050 3485	DCHN 051 1583	-2400-
-2410-	DCHN 052 556A	DCHN 053 D119	DCHN 054 C118	DCHN 055 3085	DCHN 056 5666	DCHN 057 7068	DCHN 058 7160	DCHN 059 176B	-2410-
-2420-	DCHN 060 5068	DCHN 061 5260	DCHN 062 021E	DCHN 063 0080	DCHN 064 55E9	DCHN 065 5F59	DCHN 066 54F9	DCHN 067 5B49	-2420-
-2430-	DCHN 068 D235	DCHN 069 3000	DCHN 070 2400	DCHN 071 C1C3	DCHN 072 A00E	DCHN 073 54B9	DCHN 074 55F9	DCHN 075 2135	-2430-
-2440-	DCHN 076 9C74	DCHN 090 A01E	DCHN 091 A43A	DCHN 077 5560	DCHN 078 716A	DCHN 079 7060	DCHN 080 2080	DCHN 081 A428	-2440-
-2450-	DCHN 082 5666	DCHN 083 0010	DCHN 084 1583	DCHN 085 1413	DCHN 086 7468	DCHN 087 2485	DCHN 088 2505	DCHN 089 A41E	-2450-
-2460-	CSFT 087 0002	CSFT 088 D06C	CSFT 089 611B	CSFT 090 600D	CSFT 091 633D	CSFT 092 622D	CSFT 093 677D	CSFT 094 666D	-2460-
-2470-	CSFT 095 6FFD	CSFT 096 6EED	CSFT 097 E4F8	CSFT 098 3000	CSFT 099 6AAD	CSFT 100 2DFB	CSFT 101 F0E0	CSFT 102 128E	-2470-
				248-					
-2480-	DYPE 146 1547	DYPE 147 9A90	DYPE 148 C108	DYPE 149 3485	DYPE 150 9D30	DYPE 159 3F20	DYPE 160 EB90	DYPE 161 4FFF	-2480-
-2490-	DYPE 162 9F66	DYPE 156 3523	DYPE 157 2F08	DYPE 158 9F66	DYPE 151 2EC3	DYPE 152 D560	DYPE 153 A566	DYPE 154 9258	-2490-
-24A0-	DYPE 155 93CA	DYPE 113 8DC4	DYPE 114 54F9	DYPE 115 6664	DYPE 116 9A98	DYPE 117 9028	DYPE 118 1485	DYPE 119 1C83	-24A0-
-2480-	DYPE 120 9A92	DYPE 121 2E07	DYPE 122 5FF9	DYPE 123 C4BA	DYPE 124 AAA2	DYPE 125 1D45	DYPE 126 1F28	DYPE 127 EEC2	-2480-
-24C0-	DYPE 128 3D45	DYPE 129 0F4B	DYPE 130 C4E5	DYPE 131 EEDD	DYPE 132 0F3B	DYPE 133 C499	DYPE 134 2D07	DYPE 135 0F1B	-24C0-
-24D0-	DYPE 136 C495	DYPE 137 0F9B	DYPE 138 C493	DYPE 139 0FAB	DYPE 140 C48B	DYPE 141 3D85	DYPE 142 2E23	DYPE 143 15C5	-24D0-
-24E0-	DYPE 144 F200	DYPE 145 9F80	DYPE 163 3C83	DYPE 164 9A92	DYPE 165 5DF9	DYPE 166 8F68	CDMD 203 5773	CDMD 204 0769	-24E0-
-24F0-	CDMD 205 F4F5	CDMD 206 A520	CDMD 207 F77C	CDMD 208 0719	CDMD 209 F4FD	CDMD 210 143D	CDMD 211 128E	5007	-24F0-

CONTROL ADDRESS

24--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2500-	CDMD 077 7D30	CDMD 078 7EA3	CDMD 079 C4CF	CDMD 080 9AEE	CDMD 081 0D9D	CDMD 082 E08E	CDMD 083 2040	CDMD 084 5D30	-2500-
-2510-	CDMD 085 5D7B	CDMD 086 7D9F	CDMD 087 47DB	CDMD 088 7D30	CDMD 089 0186	CDMD 090 A53C	CDMD 052 9AEE	CDMD 053 C1A4	-2510-
-2520-	CDMD 054 9254	CDMD 055 B07C	CDMD 056 F580	CDMD 057 1002	CDMD 058 9AF0	CDMD 059 9254	CDMD 060 28B3	CDMD 061 9C70	-2520-
-2530-	CDMD 062 56A2	CDMD 063 5D70	CDMD 064 5DDB	CDMD 065 7D70	CDMD 066 2993	CDMD 067 5981	CDMD 068 5004	CDMD 069 1042	-2530-
-2540-	CDMD 070 56E2	CDMD 071 9AF2	CDMD 072 5D30	CDMD 073 7D8F	CDMD 074 F081	CDMD 075 7D30	CDMD 076 A53E	CDMD 091 C1A1	-2540-
-2550-	CDMD 092 5CC2	CDMD 093 2FC3	CDMD 094 C958	CDMD 095 1F68	CDMD 096 E45D	CDMD 097 3F13	CDMD 098 5730	CDMD 099 67F5	-2550-
-2560-	DCOM 100 7730	DCOM 101 6443	DCOM 102 8BC2	DCOM 048 2E07	DCOM 049 7468	DCOM 050 7E68	DCOM 051 7060	DCOM 052 C573	-2560-
-2570-	DCOM 053 2A07	DCOM 054 128E	CFCO 143 5E18	CFCO 144 5818	CFCO 145 F47F	CFCO 146 5218	CFCO 147 5010	CFCO 148 128E	-2570-
258-									
-2580-	CDMD 047 A51C	CDMD 002 E187	CDMD 003 1445	CDMD 004 C10A	CDMD 005 AE98	CDMD 006 5575	CDMD 007 77E3	CDMD 008 E088	-2580-
-2590-	CDMD 009 8DC0	CDMD 010 5730	CDMD 011 A4EC	CDMD 012 5710	CDMD 013 3415	CDMD 014 A4EC	CDMD 015 10CE	CDMD 016 55BD	-2590-
-25A0-	CDMD 017 3883	CDMD 018 3815	CDMD 019 7AE2	CDMD 020 4CA6	CDMD 021 3825	CDMD 022 7AF2	CDMD 023 5F3A	CDMD 024 5FFB	-25A0-
-25B0-	CDMD 025 7FAF	CDMD 026 7FCA	CDMD 027 5FF1	CDMD 028 C946	CDMD 029 573A	CDMD 030 47F3	CDMD 031 77AF	CDMD 032 7FAA	-25B0-
-25C0-	CDMD 033 77CA	CDMD 034 57F5	CDMD 035 C939	CDMD 036 7FAA	CDMD 037 55E5	CDMD 038 55FD	CDMD 039 7EF1	CDMD 040 2002	-25C0-
-25D0-	CDMD 041 71E9	CDMD 042 70A9	CDMD 043 4206	CDMD 045 F001	CDMD 046 8882	DCLL 002 42A6	DCLL 003 44A6	DCLL 004 2423	-25D0-
-25E0-	DCLL 005 74B8	DCLL 006 7088	DCLL 007 2465	DCLL 008 7488	DCLL 009 2115	DCLL 010 3183	DCLL 011 7088	DCLL 012 3845	-25E0-
-25F0-	DCLL 013 46A6	DCLL 014 17FD	DCLL 015 8ADA	DCLL 016 10F5	DCLL 017 24C3	DCLL 018 87AE	FINT 002 7CC2	FINT 003 5F59	-25F0-

CONTROL ADDRESS

25--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2600-	FINT 004 D98E	FINT 005 5A32	FINT 006 2A13	FINT 007 7A32	FINT 008 8420	FINT 009 A024	FINT 010 A64C	FINT 011 C996	-2600-
-2610-	FINT 012 2E07	FINT 013 8428	FINT 014 A60A	FINT 015 2B43	FINT 016 4FBF	FINT 017 5AA0	FINT 018 2A13	FINT 019 7A32	-2610-
-2620-	FINT 020 E1A8	FINT 021 0D02	FINT 022 2E25	FINT 023 A644	FINT 024 E5AF	FINT 025 948E	FINT 026 A644	FINT 027 AFEA	-2620-
-2630-	FINT 028 5DDD	FINT 029 588B	FINT 030 6BD5	FINT 031 7A32	FINT 032 FABF	FINT 033 8224	FINT 034 A642	FINT 035 8224	-2630-
-2640-	FINT 036 9496	FINT 037 3E43	FINT 038 2F07	FINT 039 2400	FINT 040 2A07	FINT 041 A01E	FINT 042 5A32	FINT 043 2A13	-2640-
-2650-	FINT 044 1000	FINT 045 5CC2	FINT 046 A43C	CFCO 007 F45C	CFCO 008 4953	CFCO 009 F46F	CFCO 010 4153	CFCO 011 5115	-2650-
-2660-	CFCO 012 4013	CFCO 013 5005	CFCO 014 4303	CFCO 015 5335	CFCO 016 4233	CFCO 017 5225	CFCO 018 4923	CFCO 019 5995	-2660-
-2670-	CFCO 020 4893	CFCO 021 5885	CFCO 022 4F83	CFCO 023 5FF5	CFCO 024 4AF3	CFCO 025 2CFF	CFCO 026 C4D6	CFCO 027 128E	-2670-
268-									
-2680-	ALDP 248 D648	ALDP 249 OFFD	ALDP 250 E095	ALDP 251 1FCD	ALDP 252 E08F	ALDP 253 1F4D	ALDP 254 E0B6	ALDP 255 F0B7	-2680-
-2690-	ALDP 256 2F9B	ALDP 257 E0B6	ALDP 258 5F4D	ALDP 259 A6C8	ALDP 232 5A4F	ALDP 233 3683	ALDP 234 CAAF	ALDP 235 14C5	-2690-
-26A0-	ALDP 236 DAB2	ALDP 237 04AB	ALDP 238 C4AF	ALDP 239 1623	ALDP 240 045B	ALDP 241 C4B2	ALDP 242 C632	ALDP 243 1F28	-26A0-
-26B0-	ALDP 244 961C	ALDP 245 B17A	ALDP 246 E201	ALDP 247 962A	ALDP 216 3BE9	ALDP 217 56AA	ALDP 218 52AA	ALDP 219 54AA	-26B0-
-26C0-	ALDP 220 42AF	ALDP 221 C218	ALDP 222 D349	ALDP 223 4F3F	ALDP 224 2440	ALDP 225 5872	ALDP 226 2404	ALDP 227 50A0	-26C0-
-26D0-	ALDP 228 2B2D	ALDP 229 5EA0	ALDP 230 1F28	ALDP 231 128E	CFHA 044 6DDD	CFHA 045 655B	CFHA 046 F467	CFHA 047 611D	-26D0-
-26E0-	CFHA 048 600D	CFHA 049 633D	CFHA 050 622D	CFHA 051 699D	CFHA 052 688D	CFHA 053 6FFD	CFHA 054 C1F3	CFHA 055 6AAD	-26E0-
-26F0-	CFHA 056 128E	CFHA 057 6EED	CFHA 058 128E	CDVD 046 2C1B	CDVD 047 611D	CDVD 048 600D	CDVD 049 633D	CDVD 050 622D	-26F0-

CONTROL ADDRESS

26--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2700-	CDVD 051 EC0D	CDVD 052 699D	CDVD 053 688D	CDVD 054 655D	CDVD 055 644D	CDVD 056 128E	CDVD 057 C418	CDVD 058 1002	-2700-
-2710-	CDVD 059 79FD	CDVD 060 78ED	CDVD 061 757D	CDVD 062 746D	CDVD 063 5BC9	CDVD 064 52D9	CDVD 065 E827	CDVD 066 C436	-2710-
-2720-	CDVD 067 97EC	CDVD 068 C436	CDVD 069 AEFC	CDVD 070 C42B	CDVD 071 97EC	CDVD 072 F839	CDVD 073 97F0	CDVD 074 52D9	-2720-
-2730-	CDVD 075 D5BA	CDVD 076 CD3B	CDVD 077 AEFC	CDVD 078 F82D	CDVD 079 CD25	CDVD 080 58BB	CDVD 081 72A8	CDVD 082 70AA	-2730-
-2740-	CDVD 083 1B15	CDVD 084 74A8	CDVD 085 78AA	CDVD 086 9256	CDVD 087 A8DE	CDMD 184 5C79	CDMD 185 4286	CDMD 186 5D30	-2740-
-2750-	CDMD 187 5F6A	CDMD 188 7DFF	CDMD 189 7D3A	CDMD 190 C34F	CDMD 191 7BAF	CDMD 192 0002	CDMD 193 C5E2	CDMD 194 0B9F	-2750-
-2760-	CDMD 195 2002	CDMD 196 D1E8	CDMD 197 E0CA	CDMD 198 B1CC	CDMD 199 FOCA	CDMD 200 5710	CDMD 201 47BB	CDMD 202 B1C2	-2760-
-2770-	CCOM 045 5218	CCOM 046 5010	CCOM 047 55BB	CCOM 048 56A8	CCOM 049 5EAA	CCOM 050 100E	CCOM 051 128E	5007	-2770-
278-									
-2780-	DCLC 191 A162	DCLC 150 C386	DCLC 151 8760	DCLC 152 F22B	DCLC 153 2888	DCLC 154 5429	DCLC 155 C110	DCLC 156 3385	-2780-
-2790-	DCLC 157 7222	DCLC 158 2800	DCLC 159 2007	DCLC 160 7032	DCLC 161 0D04	DCLC 162 E629	DCLC 163 C9A2	DCLC 164 262B	-2790-
-27A0-	DCLC 165 C99B	DCLC 166 2400	DCLC 167 5E22	DCLC 168 A030	DCLC 174 A254	DCLC 169 5429	DCLC 170 5349	DCLC 171 ECC5	-27A0-
-27B0-	DCLC 172 D8C4	DCLC 173 A7B4	DCLC 183 2802	DCLC 184 1635	DCLC 185 E629	DCLC 186 C9C0	DCLC 187 262B	DCLC 188 C9B9	-27B0-
-27C0-	DCLC 189 E680	DCLC 190 0244	DCLC 175 1635	DCLC 176 7602	DCLC 177 3413	DCLC 178 2D40	DCLC 179 ECB4	DCLC 180 2B08	-27C0-
-27D0-	DCLC 181 3413	DCLC 182 A7B8	CFCO 120 3BE3	CFCO 121 F45C	CFCO 122 1B43	CFCO 123 F469	CFCO 124 56AA	CFCO 125 717D	-27D0-
-27E0-	CFCO 126 706D	CFCO 127 56AA	CFCO 128 737D	CFCO 129 726D	CFCO 130 56AA	CFCO 131 797D	CFCO 132 786D	CFCO 133 56A0	-27E0-
-27F0-	CFCO 134 7F7D	CFCO 135 C1F7	CFCO 136 128E	CFCO 137 7E6D	CFCO 138 128E	CCOM 181 5682	CCOM 182 427F	CCOM 183 128E	-27F0-

CONTROL ADDRESS

27--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2800-	CFCY 052 B384	CFCY 053 A150	CFCY 054 A154	CFCY 055 A152	CFCY 056 B23E	CFCY 057 B376	CFCY 058 B376	CFCY 059 B376	-2800-
-2810-	CFCY 060 B266	CFCY 061 8972	CFCY 062 8972	CFCY 063 8972	CFCY 064 9D80	CFCY 065 9D7E	CFCY 066 8972	CFCY 067 8972	-2810-
-2820-	CFCY 034 AF4E	CFCY 035 2B83	CFCY 036 8DC2	CFCY 037 0565	CFCY 038 E0C2	CFCY 039 AB53	CFCY 040 F43F	CFCY 041 0183	-2820-
-2830-	CFCY 042 FOC2	CFCY 043 C038	CFCY 044 A574	CFCY 045 A854	CFCY 046 04F5	CFCY 047 F08A	CFCY 048 8D6C	CFCY 049 01C3	-2830-
-2840-	CFCY 050 F0B3	CFCY 051 AE98	CFCY 012 E1C9	CFCY 013 3485	CFCY 014 0567	CFCY 015 C4C2	CFCY 016 8DC2	CFCY 017 55B3	-2840-
-2850-	CFCY 018 3B83	CFCY 019 8212	CFCY 021 45BB	CFCY 022 2507	CFCY 023 7AE2	CFCY 024 10CE	CFCY 025 B001	5007	-2850-
-2860-	CICY 016 B063	CICY 017 B043	CICY 018 A844	CICY 019 A844	CICY 020 88DA	CICY 021 AE90	CICY 022 A820	CICY 023 A820	-2860-
-2870-	CICY 024 9012	CICY 025 8106	CICY 026 A058	CICY 027 A058	CICY 028 A056	CICY 029 88D6	CICY 030 A056	CICY 031 8AB8	-2870-
288-									
-2880-	CBRC 012 C507	CBRC 013 81B3	CBRC 014 E502	CBRC 015 4806	CBRC 016 9CBA	CBRC 017 D507	CBRC 018 81B3	CBRC 019 F507	-2880-
-2890-	CBRC 020 81B3	CBRC 023 0040	CBRC 024 55B3	CBRC 025 C49F	CBRC 026 5EA8	CBRC 027 50A0	CBRC 031 2040	CBRC 032 5C2B	-2890-
-28A0-	CBRC 033 3245	CBRC 034 E1B6	CBRC 035 D428	CBRC 036 224D	CBRC 037 4D25	CBRC 038 2307	CBRC 039 55BB	CBRC 040 72A8	-28A0-
-28B0-	CBRC 041 78A0	CBRC 042 D187	CBRC 043 81B3	CBRC 044 5892	CBRC 045 A8A6	CBRC 005 55B3	CBRC 006 C483	CBRC 007 5EA8	-28B0-
-28C0-	CBRC 008 50A0	CBRC 011 9C11	CBIN 018 2C55	CBIN 019 81B3	CBIN 008 A770	CBIN 009 8174	CBIN 010 E4D0	CBIN 011 168D	-28C0-
-28D0-	CBIN 012 E5D7	CBIN 013 2C07	CBIN 014 81B3	CBIN 015 C645	CBIN 016 2C25	CBIN 017 81B3	CICY 003 2C07	CICY 004 81B3	-28D0-
-28E0-	CICY 179 AED0	CICY 180 A05A	CICY 181 A05A	CICY 182 A05A	CICY 183 A908	CICY 184 AC9A	CICY 185 B336	CICY 186 B32A	-28E0-
-28F0-	CICY 187 AEC4	CICY 188 A8C8	CICY 189 AC18	CICY 190 810A	CICY 191 8784	CICY 192 A218	CICY 193 ADC6	CICY 194 AF72	-28F0-

CONTROL ADDRESS

28--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2900-	CMPU 016 8023	CMPU 017 A932	CMPU 018 81B3	CMPU 019 81B3	CLOG 073 A770	CLOG 074 6F17	CLOG 075 6E07	CLOG 076 6737	-2900-
-2910-	CLOG 077 6627	CLOG 078 76A8	CLOG 079 7EAA	CLOG 080 E59B	CLOG 081 ACA8	CLJG 082 2C55	CLOG 083 81B3	CLOG 069 8A70	-2910-
-2920-	CLOG 070 A90A	CMPU 020 81A6	CMPU 021 AE12	CMPU 022 5B3A	CMPU 023 4B7D	CMPU 024 771A	CMPU 025 351D	CMPU 026 E087	-2920-
-2930-	CMPU 027 4B75	CMPU 028 771A	CMPU 029 577B	CMPU 030 25FD	CMPU 031 8083	CLOG 004 AB53	CLOG 005 7510	CLOG 006 81B3	-2930-
-2940-	CICY 147 AC34	CICY 148 AC28	CICY 149 AC0C	CICY 150 AC00	CICY 151 A91E	CICY 152 B36A	CICY 153 B372	CICY 154 B37A	-2940-
-2950-	CICY 155 AE9C	CICY 156 9148	CICY 157 B366	CICY 158 81F4	CICY 159 878C	CICY 160 99F8	CICY 161 B380	CICY 162 B36E	-2950-
-2960-	CICY 122 A05A	CICY 123 A05A	CICY 124 A05A	CICY 125 A05A	CICY 126 AE36	CICY 127 A892	CICY 128 ABCE	CICY 129 A88A	-2960-
-2970-	CICY 130 AB8E	CICY 131 AB8E	CICY 132 98F6	CICY 133 A05A	CICY 142 A05A	CICY 143 A05A	CICY 144 A05A	CICY 145 A05A	-2970-
298-									
-2980-	CTRT 022 2B33	CTRT 023 2B2D	CTRT 024 7FA0	CTRT 025 2BFD	CTRT 026 70AA	CTRT 027 2707	CTRT 028 77A0	CTRT 029 25FF	-2980-
-2990-	CTRT 030 F495	CTRT 031 ACA4	CTRT 032 AC84	CLOG 198 55BB	CLOG 199 2E07	CLOG 200 7EA8	CLOG 201 70A0	CLOG 202 81B3	-2990-
-29A0-	CICY 005 8B7A	CICY 006 AEF6	CICY 007 87C8	CICY 008 5498	CICY 009 B461	CLOG 098 6F75	CLOG 099 AB53	CLOG 100 10CE	-29A0-
-2980-	CLOG 101 5738	CLOG 102 5F10	CLOG 103 67F7	CLOG 104 7718	CLOG 105 25FF	CLOG 106 F4B1	CLOG 107 A916	5007	-2980-
-29C0-	CICY 045 2B53	CICY 046 9C70	CNVR 005 AB53	CNVR 006 0183	CNVR 007 FOCD	CNVR 008 AE98	CNVR 009 8DC2	CNVR 010 10CE	-29C0-
-29D0-	CNVR 011 F054	CNVR 012 AAFA	CNVR 013 ADB8	CNVR 014 2B07	CNVR 015 7A18	CNVR 016 C660	CNVR 017 8AE0	CNVR 018 2040	-29D0-
-29E0-	CNVR 019 42A6	CNVR 020 44A6	CNVR 021 48A6	CNVR 022 2D13	CNVR 023 C66F	CNVR 024 AD4E	CNVR 025 C66A	CNVR 026 AD3C	-29E0-
-29F0-	CNVR 027 C66F	CNVR 028 AD3E	CNVR 029 A9F0	5007	CTRT 033 AB53	CTRT 034 14A3	CTRT 035 5569	CTRT 036 25F5	-29F0-

CONTROL ADDRESS

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2A00-	CTRT 037 C904	CTRT 038 2555	CTRT 039 5B10	CTRT 040 5BF9	CTRT 041 0F2D	CTRT 042 E096	CTRT 043 F0B9	CTRT 044 0F1B	-2A00-
-2A10-	CTRT 045 F0B7	CTRT 046 0F2B	CTRT 047 F0B1	CTRT 048 E01B	CTRT 049 7B10	CTRT 050 5004	CTRT 051 26FF	CTRT 052 F4A4	-2A10-
-2A20-	CTRT 053 5F10	CTRT 054 AA08	CTRT 058 E02D	CTRT 059 E42B	CTRT 060 A8DC	CTRT 063 ACA4	CTRT 061 E428	CTRT 062 AC84	-2A20-
-2A30-	CTRT 055 1427	CTRT 056 3443	CTRT 057 AA18	CTRT 064 1443	CTRT 065 C03F	CTRT 066 A353	CTRT 067 5738	CTRT 068 4755	-2A30-
-2A40-	CTRT 069 F0E3	CTRT 070 0769	CTRT 071 F4E1	CTRT 072 E05B	CTRT 073 F05A	CTRT 074 7EE2	CTRT 075 5BE9	CTRT 076 2F07	-2A40-
-2A50-	CTRT 077 2B17	CTRT 078 7FA8	CTRT 079 70A0	CTRT 080 5EB9	CTRT 081 5EE2	CTRT 082 3427	CTRT 083 7518	CTRT 084 AA6A	-2A50-
-2A60-	CTRT 085 B07C	CTRT 086 E05D	CTRT 087 7B18	CTRT 088 D06B	CTRT 089 3463	CTRT 090 5773	CTRT 091 0769	CTRT 092 F4F5	-2A60-
-2A70-	CTRT 093 148B	CTRT 094 AA1C	CTRT 095 F77A	CTRT 096 07FD	CTRT 097 C4FC	CTRT 098 1423	CTRT 099 2707	CTRT 100 3443	-2A70-
2A8-									
-2A80-	CTRT 101 AA1C	BSWI 139 9CC6	BSWI 051 DFE8	BSWI 052 2040	BSWI 053 2080	BSWI 054 2406	BSWI 055 E990	BSWI 056 9EBA	-2A80-
-2A90-	BSWI 057 2402	BSWI 058 EB96	BSWI 059 9EEA	BSWI 060 2404	BSWI 061 EA9C	BSWI 062 9F10	BSWI 063 2482	BSWI 064 FAA2	-2A90-
-2AA0-	BSWI 065 9D6A	BSWI 066 2400	BSWI 067 2A07	BSWI 069 A7FA	BSWI 071 E1AD	BSWI 072 9258	BSWI 073 C1E3	BSWI 074 81B3	-2AA0-
-2AB0-	BSWI 091 EBB5	BSWI 093 A2FA	BSWI 099 FBB9	BSWI 100 5007	BSWI 101 EFBD	BSWI 102 A3F2	BSWI 104 DBC8	BSWI 105 DF89	-2AB0-
-2AC0-	BSWI 113 CBCC	BSWI 114 2F07	BSWI 115 508F	BSWI 116 519F	BSWI 117 0602	BSWI 118 8F06	BSWI 119 CFBE	BSWI 120 2404	-2AC0-
-2AD0-	BSWI 121 FFDC	BSWI 122 3BE9	BSWI 123 56AA	BSWI 124 D25A	BSWI 125 8734	BSWI 126 1F2A	BSWI 127 2400	BSWI 128 E903	-2AD0-
-2AE0-	BSWI 135 9CBE	BSWI 075 B354	BSWI 076 D1BE	BSWI 077 0040	BSWI 078 FFB1	BSWI 082 59BF	BSWI 083 1BFF	BSWI 088 0640	-2AE0-
-2AF0-	BSWI 089 2115	BSWI 090 9C74	CCOM 165 2400	CCOM 166 2C75	CCOM 167 81B3	CNVR 058 2B83	CNVR 059 3B15	CNVR 060 4EA6	-2AF0-

CONTROL ADDRESS

2A--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2E00-	CLOG 131 5738	CLOG 132 5F10	CLOG 133 67F5	CLOG 134 7718	CLOG 135 25FF	CLOG 136 F481	CLOG 137 A916	CMPU 045 5F7D	-2E00-
-2E10-	CMPU 046 A932	CMPU 037 5F3A	CMPU 038 25FF	CMPU 039 F08F	CMPU 040 5B3A	CMPU 041 4BF3	CMPU 042 7F1A	CMPU 043 351D	-2E10-
-2E20-	CMPU 044 8083	CLOG 086 AB53	CLOG 087 5F10	CLOG 088 6F57	CLOG 089 C4B1	CLOG 090 7F10	CLOG 091 2C55	CLOG 092 81B3	-2E20-
-2E30-	CLOG 093 7F10	CLOG 094 2C07	CLOG 095 81B3	CSTS 082 55BB	CSTS 083 5EA0	CSTS 084 4ED3	CSTS 085 5ECB	CSTS 086 1CC5	-2E30-
-2E40-	CSTS 087 FC44	CSTS 088 3C45	CSTS 089 7CC2	CSTS 090 81B3	CLOG 016 6F75	CLOG 017 AB53	CLOG 018 5738	CLOG 019 5F10	-2E40-
-2E50-	CLOG 020 4F7B	CLOG 021 7718	CLOG 022 25FF	CLOG 023 F4CD	CLOG 024 81B3	CLOG 026 6F75	CLOG 027 AB53	CLOG 028 5738	-2E50-
-2E60-	CLOG 029 5F10	CLOG 030 4F7D	CLOG 031 7718	CLOG 032 25FF	CLOG 033 F4DF	CLOG 034 81B3	CSTS 071 AB53	CSTS 072 2FF7	-2E60-
-2E70-	CSTS 073 5710	CSTS 074 7F10	CSTS 075 C77A	CSTS 076 2C55	CSTS 077 81B3	CSTS 078 2C05	CSTS 079 81B3	CMPU 008 27F5	-2E70-
				2E8-					
-2E80-	CMPU 009 C904	CMPU 010 2755	CMPU 011 5B3A	CMPU 012 5BF1	CMPU 013 7F1A	CMPU 014 5559	CMPU 015 8083	CICY 061 8071	-2E80-
-2E90-	CICY 055 AF4E	CICY 056 AB53	CICY 057 01C3	CICY 058 F08F	CICY 059 2B63	CICY 060 9C70	CLST 046 55B3	CLST 047 56A8	-2E90-
-2EA0-	CLST 048 5EAA	CLST 049 55BB	CLST 050 76A8	CLST 051 7EAA	CLST 052 81B3	CLOG 008 6F75	CLOG 009 AB53	CLOG 010 5738	-2EA0-
-2EB0-	CLOG 011 7718	CLOG 012 25FF	CLOG 013 F4AF	CLOG 014 81B3	CLOG 190 AB53	CLOG 191 55BB	CLOG 192 2B3B	CLOG 193 57A0	-2EB0-
-2EC0-	CLOG 194 7710	CLOG 195 81B3	CLST 086 55BB	CLST 087 5618	CLST 088 5E10	CLST 089 76A8	CLST 090 7EAO	CLST 091 81B3	-2EC0-
-2ED0-	CLST 078 55BB	CLST 079 56A8	CLST 080 5EAA	CLST 081 7618	CLST 082 7E10	CLST 083 81B3	CLOG 182 AB53	CLOG 183 55BB	-2ED0-
-2EE0-	CLOG 184 2B3B	CLOG 185 5710	CLOG 186 77A0	CLOG 187 81B3	CLST 094 AD26	CLST 095 52A8	CLST 096 7218	CLST 097 52A0	-2EE0-
-2EF0-	CLST 098 7218	CLST 099 AD30	CLST 100 AEEA	CICY 010 5892	CICY 011 2020	CICY 012 A9A6	CDVD 098 9256	CDVD 099 AB62	-2EF0-

CONTROL ADDRESS

2E--

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-2F00-	DCLT 008 AC74	DCLT 003 C609	DCLT 004 2007	DCLT 005 A344	DCLT 009 24F6	DCLT 010 EC80	DCLT 011 F000	DCLT 012 2400	-2F00-
-2F10-	DCLT 013 5202	DCLT 014 6311	DCLT 015 C480	DCLT 016 24F6	DCLT 017 C11C	DCLT 018 3485	DCLT 019 70F2	DCLT 020 5419	-2F10-
-2F20-	DCLT 021 5209	DCLT 022 4452	DCLT 023 7432	DCLT 024 0D04	DCLT 025 70E2	DCLT 026 0D40	DCLT 027 021E	DCLT 028 2400	-2F20-
-2F30-	DCLT 029 4652	DCLT 031 2887	DCLT 032 52AC	DCLT 033 526F	DCLT 034 5668	DCLT 036 5212	DCLT 037 50F2	DCLT 038 5EE2	-2F30-
-2F40-	DCLT 039 1F13	DCLT 040 A024	DCLT 041 B354	DCLT 042 5A22	DCLT 043 2A13	DCLT 044 2400	DCLT 045 A43E	CICY 081 5098	-2F40-
-2F50-	CICY 082 50BB	CICY 083 C4EF	CICY 084 5EA8	CICY 085 56A0	CICY 086 617B	CICY 087 606F	CICY 088 6FAD	CICY 089 55B3	-2F50-
-2F60-	CICY 090 C4ED	CICY 091 52A8	CICY 092 56A0	CICY 093 617B	CICY 094 606D	CICY 095 6F3D	CICY 096 128E	CICY 097 2F07	-2F60-
-2F70-	CICY 098 C4DF	CBIN 080 A770	CBIN 081 8174	CBIN 082 ADCA	DMCS 009 2E85	DMCS 010 74E2	DMCS 011 7EF2	DMCS 012 2F10	-2F70-
									2F8-
-2F80-	DMCS 013 EBOF	DMCS 014 8612	DMCS 003 52A2	DMCS 004 6DD3	DMCS 005 DD16	DMCS 006 2FF5	DMCS 007 AFB0	DMCS 018 2D13	-2F80-
-2F90-	DMCS 019 72A2	DMCS 020 2427	DMCS 021 25A3	DMCS 022 4F48	DMCS 023 5FC9	DMCS 024 4826	DMCS 025 69FB	DMCS 026 F5A2	-2F90-
-2FA0-	DMCS 027 281B	DMCS 028 5838	DMCS 029 5198	DMCS 030 2C1B	DMCS 031 6B11	DMCS 032 C484	DMCS 033 OCFD	DMCS 034 C4A2	-2FA0-
-2FB0-	DMCS 035 49DF	DMCS 036 27FF	DMCS 037 7D60	DMCS 038 52A2	DMCS 043 2C75	DMCS 047 27F7	DMCS 048 77C8	DMCS 049 E83C	-2FB0-
-2FC0-	DMCS 050 2105	DMCS 051 71C0	DMCS 052 B2C6	FILE 296 2E07	FILE 297 5C32	FILE 298 0E08	FILE 299 C965	FILE 300 D965	-2FC0-
-2FD0-	FILE 301 2C85	FILE 302 E956	FILE 303 2C25	FILE 304 F95E	FILE 305 2C45	FILE 306 E95E	FILE 307 2C15	FILE 308 2E07	-2FD0-
-2FE0-	FILE 309 4BEF	FILE 310 49CF	FILE 311 4EEF	FILE 312 4DEF	FILE 313 128E	FILE 280 2D95	FILE 281 0E08	FILE 282 2100	-2FE0-
-2FF0-	FILE 283 CCD1	FILE 284 2D1B	FILE 285 DCD9	FILE 286 2D1B	FILE 287 ECD5	FILE 288 2D1B	FILE 289 FCDD	FILE 290 128E	-2FF0-

CONTROL ADDRESS

2F--

ADDRESS LIST

DATE 11/08/68

30—

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-3000-	BCHK 082 A16C	BCHK 064 4252	BCHK 065 4C26	BCHK 066 4AC0	BCHK 067 2DEB	BCHK 068 8F2C	BCHK 069 6CC4	BCHK 070 DC0A	-3000-
-3010-	BCHK 071 42A6	BCHK 072 C481	BCHK 073 2CE3	BCHK 074 2DC5	BCHK 075 3D63	BCHK 076 C99F	BCHK 077 2820	BCHK 079 8F2C	-3010-
-3020-	BCHK 080 6AC0	BCHK 081 B02C	BCHK 005 2404	BCHK 045 4632	BCHK 046 6662	BCHK 047 FA82	BCHK 048 21C5	BCHK 049 20E3	-3020-
-3030-	BCHK 050 6004	BCHK 051 4208	BCHK 052 2A07	BCHK 053 538D	BCHK 054 C48C	BCHK 055 2B1B	BCHK 056 68B3	BCHK 057 42A0	-3030-
-3040-	BCHK 058 6208	BCHK 059 0610	BCHK 060 511B	BCHK 061 2723	BCHK 062 20C0	BCHK 063 968A	CSFT 107 D05E	CSFT 108 5115	-3040-
-3050-	CSFT 109 4013	CSFT 110 5005	CSFT 111 4303	CSFT 112 5335	CSFT 113 4233	CSFT 114 5225	CSFT 115 4723	CSFT 116 5775	-3050-
-3060-	CSFT 117 4673	CSFT 118 5665	CSFT 119 4F63	CSFT 120 5FF5	CSFT 121 4EF3	CSFT 122 5EE5	CSFT 123 4AE3	CSFT 124 128E	-3060-
-3070-	CMLT 159 3002	CMLT 160 9CD2	CMLT 161 1002	CMLT 162 3D25	CMLT 163 9CD4	CMLT 164 8344	CCOM 071 2B73	CCOM 072 9C70	-3070-
				308-					
-3080-	DCLR 057 0240	DCLR 002 C3C2	DCLR 003 875A	DCLR 013 D623	DCLR 017 1773	DCLR 028 A078	DCLR 029 C510	DCLR 030 2EC3	-3080-
-3090-	DCLR 031 15C5	DCLR 032 34A5	DCLR 033 C398	DCLR 034 9EE0	DCLR 035 E71E	DCLR 036 F71E	DCLR 037 83A0	DCLR 038 B16A	-3090-
-30A0-	DCLR 039 9D52	DCLR 041 2425	DCLR 042 54E9	DCLR 043 15C5	DCLR 044 1573	DCLR 045 CCB0	DCLR 046 2B00	DCLR 047 945A	-30A0-
-30B0-	DCLR 048 0D10	DCLR 049 3413	DCLR 050 2205	DCLR 051 7602	DCLR 052 2D40	DCLR 053 C681	DCLR 054 E681	DCLR 055 2B00	-30B0-
-30C0-	DCLR 056 A7B6	DCLR 004 F207	DCLR 008 2400	DCLR 009 2F25	DCLR 010 2E05	DCLR 011 A030	DYPE 071 2F53	DYPE 072 3F15	-30C0-
-30D0-	DYPE 073 1F00	DYPE 074 2F08	DYPE 075 4FFF	DYPE 076 0F20	DYPE 077 3485	DYPE 078 1483	DYPE 079 5DD9	DYPE 080 C4E5	-30D0-
-30E0-	DYPE 081 3E23	DYPE 082 15C5	DYPE 083 D56B	DYPE 084 3E83	DYPE 085 B10A	DYPE 086 9A90	DYPE 087 AAA2	CMLT 148 3002	-30E0-
-30F0-	CMLT 149 9CD2	CMLT 150 1002	CMLT 151 3D25	CMLT 152 9CD2	CMLT 153 8344	DCHN 097 2400	DCHN 098 2C75	DCHN 099 A164	-30F0-

CONTROL ADDRESS

30—

ADDRESS LIST

DATE 11/08/68

31--

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-3100-	DCOM 041 C804	DCOM 042 2F04	DCOM 043 128E	DCOM 006 C140	DCOM 007 2683	DCOM 008 F017	DCOM 009 221E	DCOM 010 C785	-3100-
-3110-	DCOM 011 2010	DCOM 012 021E	DCOM 013 3413	DCOM 014 76D2	DCOM 015 1683	DCOM 016 F727	DCOM 017 E723	DCOM 018 0D08	-3110-
-3120-	DCOM 019 128E	DCOM 023 0F20	DCOM 024 128E	DCOM 020 E72D	DCOM 021 0820	DCOM 022 128E	DCOM 025 0F24	DCOM 026 C832	-3120-
-3130-	DCOM 027 1C83	DCOM 028 128E	DCOM 029 021E	DCOM 030 1683	DCOM 031 D541	DCOM 032 F747	DCOM 033 E743	DCOM 034 2D88	-3130-
-3140-	DCOM 035 128E	DCOM 039 2F30	DCOM 040 128E	DCOM 036 E701	DCOM 037 2B30	DCOM 038 128E	DPTT 071 FED0	DPTT 072 3E13	-3140-
-3150-	DPTT 073 EED6	DPTT 074 3E23	DPTT 075 3F13	DPTT 076 C85A	DPTT 077 3E23	DPTT 078 5E3F	DPTT 079 1393	DPTT 080 FOE5	-3150-
-3160-	DPTT 081 3E23	DPTT 082 3F15	DPTT 083 3E43	DPTT 084 OD0A	DPTT 085 128E	ERRQ 110 F772	ERRQ 111 D271	ERRQ 112 OD04	-3160-
-3170-	ERRQ 113 128E	ERRQ 114 OD80	ERRQ 115 128E	CBIN 051 ADAB	CBIN 052 810C	DYPE 231 5AFF	DYPE 232 5F19	DYPE 233 11C5	-3170-
318-									
-3180-	DYPE 234 CF10	DYPE 235 2055	DYPE 236 218D	DYPE 237 218B	DYPE 238 5F00	DYPE 239 4FFF	DYPE 240 0F20	DYPE 241 128E	-3180-
-3190-	DYPE 242 C483	DYPE 243 F098	DYPE 244 11CB	DYPE 245 C482	DYPE 246 E0B1	DYPE 247 C820	DYPE 248 DB03	DYPE 249 E82D	-3190-
-31A0-	DYPE 250 011F	DYPE 251 C4A9	DYPE 252 1FBD	DYPE 253 C48A	DYPE 254 1F7D	DYPE 255 C48A	DYPE 256 111B	DYPE 258 C482	-31A0-
-31B0-	DYPE 259 CB34	DYPE 260 EB39	DYPE 261 1FFD	DYPE 262 C48A	DYPE 263 114B	DYPE 264 C482	CDMD 151 2040	CDMD 152 4826	-31B0-
-31C0-	CDMD 153 57B9	CDMD 154 C5C6	CDMD 155 0B9F	CDMD 156 008E	CDMD 157 D1DA	CDMD 158 E0EE	CDMD 159 7B1A	CDMD 160 5886	-31C0-
-31D0-	CDMD 161 5710	CDMD 162 0040	CDMD 163 2EFF	CDMD 164 C4C0	CDMD 165 8BB8	CDMD 174 FOCB	CDMD 175 58BD	CDMD 176 2A93	-31D0-
-31E0-	CDMD 177 56F2	CDMD 178 57C9	CDMD 179 1000	CDMD 180 0BB1	CDMD 181 E0ED	CDMD 182 3002	CDMD 183 A74C	CDMD 166 2040	-31E0-
-31F0-	CDMD 167 2A95	CDMD 168 56E2	CDMD 169 57C9	CDMD 170 1000	CDMD 171 0BB9	CDMD 172 F4EB	CDMD 173 A74C	FILX 088 OD83	-31F0-

31--

CONTROL ADDRESS

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-3200-	FILX 089 F084	FILX 090 EDOC	FILX 091 97A6	FILX 117 2F07	FILX 118 2EC3	FILX 119 8E3A	FILX 092 3D13	FILX 093 2C25	-3200-
-3210-	FILX 094 C916	FILX 095 FD1C	FILX 096 3DF3	FILX 097 FD1C	FILX 098 30D3	FILX 099 1D15	FILX 100 CD25	FILX 101 3D85	-3210-
-3220-	FILX 102 DD28	FILX 103 1DFD	FILX 104 DD29	FILX 105 3DA5	FILX 106 7C52	FILX 107 2F45	FILX 108 CD89	FILX 109 F9B5	-3220-
-3230-	FILX 110 4886	FILX 111 C488	FILX 112 DD86	FILX 113 2E08	FILX 114 0E04	FILX 115 5E42	FILX 116 862A	CFHA 006 2C13	-3230-
-3240-	CFHA 007 A6DC	CFHA 008 A6DC	CFHA 009 A6DC	CFHA 010 A656	CFHA 011 5CC2	CFHA 012 D5E4	CFHA 013 2A07	CFHA 014 OFF3	-3240-
-3250-	CFHA 015 C4E6	CFHA 016 A1E4	CFHA 017 2507	CFHA 018 2EFF	CFHA 019 E4F2	CFHA 020 F4F5	CFHA 021 B24E	CFHA 028 ED67	-3250-
-3260-	CFHA 029 0080	CFHA 030 8136	CFHA 031 2E07	CFHA 032 9A00	CFHA 033 925A	CFHA 034 C1EF	CFHA 035 A8DE	CFHA 036 57B9	-3260-
-3270-	CFHA 037 9C70	CFHA 022 F4DD	CFHA 023 ED62	CFHA 024 1E8D	CFHA 025 27D3	CFHA 026 2080	CFHA 027 B24E	5007	-3270-
									328-
-3280-	CSFT 140 6EA1	CSFT 141 C48C	CSFT 142 D18A	CSFT 143 CF0F	CSFT 144 3000	CSFT 145 CF0E	CSFT 146 3000	CSFT 147 5FE9	-3280-
-3290-	CSFT 148 56F9	CSFT 149 5769	CSFT 150 D031	CSFT 151 2707	CSFT 152 128E	CSFT 129 F001	CSFT 130 D026	CSFT 131 5019	-3290-
-32A0-	CSFT 132 5309	CSFT 133 5239	CSFT 134 5729	CSFT 135 5679	CSFT 136 5F69	CSFT 137 5EF9	CSFT 138 5AE9	CSFT 139 128E	-32A0-
-32B0-	CSFT 153 5279	CSFT 154 5329	CSFT 155 5039	CSFT 156 5109	CSFT 157 2107	CSFT 158 128E	DMCS 056 5D38	DMCS 057 3E43	-32B0-
-32C0-	DMCS 058 211B	DMCS 059 71C0	DMCS 060 CE4D	DMCS 061 2D4D	DMCS 062 264D	DMCS 063 F4C2	DMCS 064 CBE7	DMCS 065 CF67	-32C0-
-32D0-	DMCS 066 DA66	DMCS 070 1E43	DMCS 071 2515	DMCS 072 C167	DMCS 073 4948	DMCS 074 69D1	DMCS 075 C4D6	DMCS 076 F163	-32D0-
-32E0-	DMCS 077 6446	DMCS 078 4D40	DMCS 079 B2C2	DMCS 081 51B9	DMCS 082 6BF1	DMCS 083 C4BC	DMCS 087 54E2	DMCS 088 5EF2	-32E0-
-32F0-	DMCS 089 4642	DMCS 090 2105	DMCS 091 2E43	DMCS 092 8D2E	CCOM 173 2BB1	CCOM 174 7FA0	CCOM 175 128E	5007	-32F0-

CONTROL ADDRESS

ADDRESS LIST

DATE 11/08/68

CONTROL ADDRESS	0	2	4	6	8	A	C	E	
-3300-	CSTS 118 B35A	CSTS 104 F901	CSTS 105 5310	CSTS 106 495F	CSTS 107 473F	CSTS 108 A8DE	CSTS 109 F901	CSTS 110 495F	-3300-
-3310-	CSTS 111 3F00	CSTS 112 C092	CSTS 113 585F	CSTS 114 1F00	CSTS 115 15FF	CSTS 116 7510	CSTS 117 A8DE	DPTT 088 F024	-3310-
-3320-	DPTT 089 1413	DPTT 090 0010	DPTT 091 4046	DPTT 092 1485	DPTT 093 128E	CLOG 144 A770	CLOG 145 6F11	CLOG 146 6E01	-3320-
-3330-	CLOG 147 6731	CLOG 148 6621	CLOG 149 A912	CLOG 114 A770	CLOG 115 6F15	CLOG 116 6E05	CLOG 117 6735	CLOG 118 6625	-3330-
-3340-	CLOG 119 A912	FILE 242 2112	FILE 243 5ECF	FILE 244 6CD1	FILE 245 C4CD	FILE 246 8C7E	FILE 247 128E	CMLT 165 9CD2	-3340-
-3350-	CMLT 166 3002	CMLT 167 8076	CCOM 186 28B1	CCOM 187 5FA0	CCOM 188 128E	CICY 068 2B23	CICY 069 9C70	BWRP 090 A062	-3350-
-3360-	BWRP 091 9270	CCOM 154 2400	CCOM 155 A8DC	CBIN 026 8A70	CBIN 027 AC1A	CLOG 036 8A70	CLOG 037 AC9C	CBIN 076 8A70	-3360-
-3370-	CBIN 077 AF74	CLOG 110 8A70	CLOG 111 8338	CFCY 068 925A	CFCY 069 A05A	CLOG 140 8A70	CLOG 141 832C	5007	-3370-
338-									
-3380-	CBIN 055 8A70	CBIN 056 ADC8	CFLS 007 1E85	CFLS 008 A154	CBIN 022 ADA8	CBIN 023 ABCA	5007	5007	-3380-

ADDRESS LIST										DATE 11/08/68
AUX STORAGE	0	2	4	6	8	A	C	E		
-3000-	ERCX 075 40F1	ERCX 075 F2F3	ERCX 075 F4F5	ERCX 075 F6F7	ERCX 075 F061	ERCX 075 E2E3	ERCX 075 E4E5	ERCX 075 E6E7	-3000-	
-3010-	ERCX 075 60D1	ERCX 075 D2D3	ERCX 075 D4D5	ERCX 076 D6D7	ERCX 076 D0A1	ERCX 076 A2A3	ERCX 076 A4A5	ERCX 076 A6A7	-3010-	
-3020-	ERCX 076 50C1	ERCX 076 C2C3	ERCX 076 C4C5	ERCX 076 C6C7	ERCX 076 C081	ERCX 076 8283	ERCX 077 8485	ERCX 077 8687	-3020-	
-3030-	ERCX 077 6A91	ERCX 077 9293	ERCX 077 9495	ERCX 077 9697	ERCX 077 7081	ERCX 077 B2B3	ERCX 077 B4B5	ERCX 077 B6B7	-3030-	
-3040-	ERCX 077 F931	ERCX 078 3233	ERCX 078 3435	ERCX 078 3637	ERCX 078 E921	ERCX 078 2223	ERCX 078 2425	ERCX 078 2627	-3040-	
-3050-	ERCX 078 D911	ERCX 078 1213	ERCX 078 1415	ERCX 078 1617	ERCX 079 A9E1	ERCX 079 6263	ERCX 079 6465	ERCX 079 6667	-3050-	
-3060-	ERCX 079 C901	ERCX 079 0203	ERCX 079 0405	ERCX 079 0607	ERCX 079 8941	ERCX 079 4243	ERCX 079 4445	ERCX 080 4647	-3060-	
-3070-	ERCX 080 9951	ERCX 080 5253	ERCX 080 5455	ERCX 080 5657	ERCX 080 B971	ERCX 080 7273	ERCX 080 7475	ERCX 080 7677	-3070-	
										308-
-3080-	ERCX 080 F879	ERCX 080 7A7B	ERCX 081 7C7D	ERCX 081 7E7F	ERCX 081 E869	ERCX 081 E06B	ERCX 081 6C6D	ERCX 081 6E6F	-3080-	
-3090-	ERCX 081 D859	ERCX 081 5A5B	ERCX 081 5C5D	ERCX 081 5E5F	ERCX 081 A8A0	ERCX 082 AAAB	ERCX 082 ACAD	ERCX 082 AEAF	-3090-	
-30A0-	ERCX 082 C849	ERCX 082 4A4B	ERCX 082 4C4D	ERCX 082 4E4F	ERCX 082 8880	ERCX 082 8A8B	ERCX 082 8C8D	ERCX 082 8E8F	-30A0-	
-30B0-	ERCX 083 9890	ERCX 083 9A9B	ERCX 083 9C9D	ERCX 083 9E9F	ERCX 083 B880	ERCX 083 BABB	ERCX 083 BCBD	ERCX 083 BEBF	-30B0-	
-30C0-	ERCX 083 3839	ERCX 083 3A3B	ERCX 083 3C3D	ERCX 084 3E3F	ERCX 084 2829	ERCX 084 2A2B	ERCX 084 2C2D	ERCX 084 2E2F	-30C0-	
-30D0-	ERCX 084 1819	ERCX 084 1A1B	ERCX 084 1C1D	ERCX 084 1E1F	ERCX 084 6820	ERCX 084 EAEB	ERCX 085 ECED	ERCX 085 EEEF	-30D0-	
-30E0-	ERCX 085 0809	ERCX 085 0A0B	ERCX 085 0C0D	ERCX 085 0E0F	ERCX 085 4800	ERCX 085 CACB	ERCX 085 CCCD	ERCX 085 CECF	-30E0-	
-30F0-	ERCX 085 5810	ERCX 086 DADB	ERCX 086 DCDD	ERCX 086 DEDF	ERCX 086 7830	ERCX 086 FAFB	ERCX 086 FCFD	ERCX 086 FEFF	-30F0-	

AUX STORAGE

30--

ADDRESS LIST										DATE 11/08/68	60--
AUX STORAGE	0	2	4	6	8	A	C	E			
-6000-	EPXF 089 B989	EPXF 089 8A8B	EPXF 089 8C8D	EPXF 089 8E8F	EPXF 089 9899	EPXF 089 9A9B	EPXF 089 9C9D	EPXF 089 9E9F	-6000-		
-6010-	EPXF 089 D949	EPXF 089 4A4B	EPXF 089 4C4D	EPXF 090 4E4F	EPXF 090 5859	EPXF 090 5A5B	EPXF 090 5C5D	EPXF 090 5E5F	-6010-		
-6020-	EPXF 090 7929	EPXF 090 2A2B	EPXF 090 2C2D	EPXF 090 2E2F	EPXF 090 3839	EPXF 090 3A3B	EPXF 091 3C3D	EPXF 091 3E3F	-6020-		
-6030-	EPXF 091 F909	EPXF 091 0A0B	EPXF 091 0C0D	EPXF 091 0E0F	EPXF 091 1819	EPXF 091 1A1B	EPXF 091 1C1D	EPXF 091 1E1F	-6030-		
-6040-	EPXF 091 00A9	EPXF 092 AAAB	EPXF 092 ACAD	EPXF 092 AEAF	EPXF 092 B891	EPXF 092 9293	EPXF 092 9495	EPXF 092 9697	-6040-		
-6050-	EPXF 092 80C9	EPXF 092 CACB	EPXF 092 CCCD	EPXF 092 CECF	EPXF 093 D851	EPXF 093 5253	EPXF 093 5455	EPXF 093 5657	-6050-		
-6060-	EPXF 093 4021	EPXF 093 6A6B	EPXF 093 6C6D	EPXF 093 6E6F	EPXF 093 7831	EPXF 093 C033	EPXF 093 3435	EPXF 094 3637	-6060-		
-6070-	EPXF 094 E0E9	EPXF 094 EAEB	EPXF 094 ECED	EPXF 094 EEEE	EPXF 094 F811	EPXF 094 1213	EPXF 094 1415	EPXF 094 1617	-6070-		
										608-	
-6080-	EPXF 094 B1A1	EPXF 094 A2A3	EPXF 095 A4A5	EPXF 095 A6A7	EPXF 095 B0A8	EPXF 095 B2B3	EPXF 095 B4B5	EPXF 095 B6B7	-6080-		
-6090-	EPXF 095 D1C1	EPXF 095 C2C3	EPXF 095 C4C5	EPXF 095 C6C7	EPXF 095 D0C8	EPXF 096 D2D3	EPXF 096 D4D5	EPXF 096 D6D7	-6090-		
-60A0-	EPXF 096 7161	EPXF 096 6263	EPXF 096 6465	EPXF 096 6667	EPXF 096 7068	EPXF 096 7273	EPXF 096 7475	EPXF 096 7677	-60A0-		
-60B0-	EPXF 097 F1E1	EPXF 097 E2E3	EPXF 097 E4E5	EPXF 097 E6E7	EPXF 097 F0E8	EPXF 097 F2F3	EPXF 097 F4F5	EPXF 097 F6F7	-60B0-		
-60C0-	EPXF 097 A081	EPXF 097 8283	EPXF 097 8485	EPXF 098 8687	EPXF 098 9088	EPXF 098 BABB	EPXF 098 BCBD	EPXF 098 BEBF	-60C0-		
-60D0-	EPXF 098 6041	EPXF 098 4243	EPXF 098 4445	EPXF 098 4647	EPXF 098 5048	EPXF 098 DADB	EPXF 099 DCDD	EPXF 099 DEDF	-60D0-		
-60E0-	EPXF 099 3269	EPXF 099 2223	EPXF 099 2425	EPXF 099 2627	EPXF 099 3028	EPXF 099 7A7B	EPXF 099 7C7D	EPXF 099 7E7F	-60E0-		
-60F0-	EPXF 099 2001	EPXF 100 0203	EPXF 100 0405	EPXF 100 0607	EPXF 100 1008	EPXF 100 FAFB	EPXF 100 FCFD	EPXF 100 FEFF	-60F0-		

AUX STORAGE

60--

ADDRESS LIST		DATE 11/08/68							70--
AUX STORAGE	0	2	4	6	8	A	C	E	
-7000-	DPTC 111 0025	DPTC 111 2627	DPTC 111 2829	DPTC 111 2A2B	DPTC 111 2C2D	DPTC 111 2E2F	DPTC 111 3018	DPTC 111 2E00	-7000-
-7010-	DPTC 111 1619	DPTC 111 1A1B	DPTC 111 1C1D	DPTC 112 1E1F	DPTC 112 2021	DPTC 112 2223	DPTC 112 2430	DPTC 112 0000	-7010-
-7020-	DPTC 112 220D	DPTC 112 0E0F	DPTC 112 1011	DPTC 112 1213	DPTC 112 1415	DPTC 112 1617	DPTC 113 1800	DPTC 113 1700	-7020-
-7030-	DPTC 113 0A01	DPTC 113 0203	DPTC 113 0405	DPTC 113 0607	DPTC 113 0809	DPTC 113 0A0B	DPTC 113 0C0C	DPTC 113 0800	-7030-
-7040-	DPTC 113 0025	DPTC 114 2627	DPTC 114 2829	DPTC 114 2A2B	DPTC 114 2C2D	DPTC 114 2E2F	DPTC 114 3018	DPTC 114 2E00	-7040-
-7050-	DPTC 114 1619	DPTC 114 1A1B	DPTC 114 1C1D	DPTC 114 1E1F	DPTC 115 2021	DPTC 115 2223	DPTC 115 2430	DPTC 115 0000	-7050-
-7060-	DPTC 115 220D	DPTC 115 0E0F	DPTC 115 1011	DPTC 115 1213	DPTC 115 1415	DPTC 115 1617	DPTC 115 1800	DPTC 116 1700	-7060-
-7070-	DPTC 116 0A01	DPTC 116 0203	DPTC 116 0405	DPTC 116 0607	DPTC 116 0809	DPTC 116 0A0B	DPTC 116 0C0C	DPTC 116 0800	-7070-
708-									
-7080-	DPTC 116 1625	DPTC 116 2627	DPTC 117 2829	DPTC 117 2A2B	DPTC 117 2C2D	DPTC 117 2E2F	DPTC 117 3018	DPTC 117 2E00	-7080-
-7090-	DPTC 117 2219	DPTC 117 1A1B	DPTC 117 1C1D	DPTC 117 1E1F	DPTC 117 2021	DPTC 118 2223	DPTC 118 2430	DPTC 118 0000	-7090-
-70A0-	DPTC 118 000D	DPTC 118 0E0F	DPTC 118 1011	DPTC 118 1213	DPTC 118 1415	DPTC 118 1617	DPTC 118 1800	DPTC 118 1700	-70A0-
-70B0-	DPTC 119 0A01	DPTC 119 0203	DPTC 119 0405	DPTC 119 0607	DPTC 119 0809	DPTC 119 0A0B	DPTC 119 0C0C	DPTC 119 0800	-70B0-
-70C0-	DPTC 119 1625	DPTC 119 2627	DPTC 119 2829	DPTC 120 2A2B	DPTC 120 2C2D	DPTC 120 2E2F	DPTC 120 3018	DPTC 120 2E00	-70C0-
-70D0-	DPTC 120 2219	DPTC 120 1A1B	DPTC 120 1C1D	DPTC 120 1E1F	DPTC 120 2021	DPTC 120 2223	DPTC 121 2430	DPTC 121 0000	-70D0-
-70E0-	DPTC 121 000D	DPTC 121 0E0F	DPTC 121 1011	DPTC 121 1213	DPTC 121 1415	DPTC 121 1617	DPTC 121 1800	DPTC 121 1700	-70E0-
-70F0-	DPTC 121 0A01	DPTC 122 0203	DPTC 122 0405	DPTC 122 0607	DPTC 122 0809	DPTC 122 0A0B	DPTC 122 0C0C	DPTC 122 0800	-70F0-

****ERROR MESSAGES****

NO BOUNDARY ERRORS
NO STRING CONTROL BLOCK ERRORS

THERE ARE NO DUPLICATE ASSIGNMENTS IN THIS RUN.

SR25-5402-0

System/360 Model 25 Microprogram Listing System/360 Emulator - *E60 Supplementary Course Material Printed in U.S.A. SR25-5402-0

IBM

International Business Machines Corporation
Field Engineering Division
112 East Post Road, White Plains, N.Y. 10601