

TEXAS INSTRUMENTS

Improving Man's Effectiveness Through Electronics

Model 990 Computer Co-Resident Sys Listing

948931-0001

Digital Systems Division





TEXAS INSTRUMENTS
INCORPORATED

DATE 03/10/77

LIST OF MATERIAL

PAGE 1 of 3

PART NUMBER	REV
LM 0948931-0001	A

PRINT ITEM NUMBER	QUANTITY PER ASSEMBLY	UNIT OF ISSUE	DWG. SIZE	PART NUMBER	DESCRIPTION	VENDOR PART NUMBER
0001	00001.000	EA		0948929-9901	LML,PXRATE,CO-RES PROTO SYS MON-PXR990	
0001A					THIS KIT CONTAINS ONLY	
0001B					THOSE PARTS WHICH ARE	
0001C					UNIQUE TO PXPATE. ALL OTHER	
0001D					PARTS LINKED IN PXRATE MAY	
0001E					BE FOUND IN THE STANDARD	
0001F					PROTOTYPING SYSTEM LISTINGS	
0001G					KIT (943518-0019).	
0001H					*	
0001I					.PXR SRC.PXRLNQ=LINK CONTROL	
0001J					.PX.RATE(LL,MN,PP,AD,BF,HL,	
0001K					PH)=LINKED OBJECT	
0001L					.PX.RATE(XPLAIN) EXPLAINS	
0001M					PX.RATE FILE	
0002	00001.000	EA		0948925-9901	AL,PXRASH,ONE-PASS ASSEMBLER-PXR990	
0002A					.PXR SRC.PXRASH	
0002B					.PXROBJ.NEWOBJ(ASM)	
0003	00001.000	EA		0948926-9901	AL,PXRLAL,LINKING LOADER-PXR990	
0003A					.PXR SRC.PXRLAL	
0003B					.PXROBJ.NEWOBJ(LAL)	

DRAFTSMAN	DATE	CKD. DRAFTSMAN	DATE	DESIGN ENGINEER	DATE	TITLE
						SP,CO-RES PROTO SYS LISTINGS(AL)-PXR990
APPD.-MFG.	DATE	APPD. PROJECT ENGINEER	DATE	RELEASED	DATE	PROJECT NO.
		<i>MaMote</i>	3-11-77	<i>Approved</i>	3-10-77	7506
						PART NUMBER
						LM 0948931-0001
						REV
						A

O



O



TEXAS INSTRUMENTS
INCORPORATED

DATE 03/10/77

LIST OF MATERIAL

PAGE 2 of 3

PART NUMBER	REV
LM 0948931-0001	A

PRINT ITEM NUMBER	QUANTITY PER ASSEMBLY	UNIT OF ISSUE	DWG. SIZE	PART NUMBER	DESCRIPTION	VENDOR PART NUMBER
0004	00001.000	EA		0948927-9901	AL,PXREDT,TEXT EDITOR-PXR990	
0004A					.PXR SRC.PXREDT	
0004B					.PXROBJ.NEWOBJ(EDT)	
0005	00001.000	EA		0948928-9901	AL,PXRBUF,RES MONITOR BUFFER DEF-PXR990	
0005A					.PXR SRC.PXRBUF	
0005B					.PXROBJ.NEWOBJ(PXRBUF)	
0006	00001.000	EA		0948930-9901	AL,CMDEFQ,COMMAND DEFINITION TBL-PXR990	
0006A					.PXR SRC.CMDEFQ	
0006B					.PXROBJ.NEWOBJ(CMDEFQ)	
0007	00001.000	EA		0948934-9901	AL,PRLDDR,LOADER DRIVER-PXR990	
0007A					.PXR SRC.PRLDDR	
0007B					.PXROBJ.NEWOBJ(PRLDDR)	
0008	00001.000	EA		0948935-9901	AL,OVELAY,OVERLAY SUPERVISOR-PXR990	
0008A					.PXR SRC.OVELAY	
0008B					.PXROBJ.NEWOBJ(OVELAY)	
0009	00001.000	EA		0948937-9901	AL,PROMLD,PROM LOADER-PXR990	
0009A					.PXR SRC.PROMLD	
0009B					.PXR SRC.NEWOBJ(PROMLD)	
0010	00001.000	EA		0948938-9901	AL,MINIT,INITIALIZE MONITOR-PXR990	
0010A					.PXR SRC.INITM	

DRAFTSMAN	DATE	CKD. DRAFTSMAN	DATE	DESIGN ENGINEER	DATE	TITLE	SP,CO-RES PROTO SYS LISTINGS(ALI)-PXR990	
APPD.-MFG.	DATE	APPD. PROJECT ENGINEER	DATE	RELEASED	DATE	PROJECT NO.		
							PART NUMBER	REV
							LM 0948931-0001	A

O

C

O



TEXAS INSTRUMENTS
INCORPORATED

DATE **03/10/77**

LIST OF MATERIAL

PAGE **3 of 3**

PART NUMBER	REV
LM 0948931-0001	A

PRINT ITEM NUMBER	QUANTITY PER ASSEMBLY	UNIT OF ISSUE	DWG. SIZE	PART NUMBER	DESCRIPTION	VENDOR PART NUMBER	
0010B					.PXROBJ.NEWOBJ(MINIT)		
0011	00001.000	EA		0948939-9901	AL,MPXIDS,I/O DISPATCHER-PXR990		
0011A					.PXR SRC.IODISM		
0011B					.PXROBJ.NEWOBJ(DISIO)		
0012	00001.000	EA		0948941-9901	AL,TTYPT,PAPER TAPE DSR-PXR990		
0012A					.PXR SRC.TTYPTS		
0012B					.PXROBJ.NEWOBJ(TTYPT)		
0013	00001.000	EA		0936187-9901	AL,RUNPGN RUN USER PROGRAM-PXR990		
0013A					.PXR SRC.RUNPGN		
0013B					.PXROBJ.NEWOBJ(RUNPGN)		
0014	00001.000	EA		0936145-9901	AL,BASCON,SAMPLE PROGRAM 3-PXR990		
0014A					.PX.RATE(SMP3)		
0015	00001.000	EA		0936146-9901	AL,OPTST,SAMPLE PROGRAM 4-PXR990		
0015A					.PX.RATE(SMP4)		
DRAFTSMAN		DATE	CKD. DRAFTSMAN	DATE	DESIGN ENGINEER	DATE	TITLE
							SP,CO-RES PROTO SYS LISTINGS(AL)-PXR990
APPD.-MFG.		DATE	APPD. PROJECT ENGINEER	DATE	RELEASED	DATE	PROJECT NO.
						PART NUMBER	REV
						LM 0948931-0001	A

O

O

O

APPLICATION		REVISIONS			
NEXT ASSY	USED ON	LTR	DESCRIPTION	DATE	APPROVED
	7506				

REV																						
SHEET																						
REV STATUS OF SHEETS	REV																					
	SHEET																					

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES
 ANGLES ±1°
 3 PLACE DECIMAL ± 010
 2 PLACE DECIMAL ± 02

IDENTIFYING NUMBERS SHOWN IN PARENTHESES FOR REFERENCE ONLY

INTERPRET DWG IN ACCORDANCE WITH MIL-STD-100

DWN _____ DATE _____

CHK _____

ENGR _____

QA _____

APVD _____

CONTR NO _____

DESIGN ACTIVITY RELEASE _____

 **TEXAS INSTRUMENTS**
 INCORPORATED
 Equipment Group Dallas, Texas

LML, PXRATE, CO-RESIDENT PROTOTYPING SYSTEM DEBUG MONITOR-PXR990

SIZE A	CODE IDENT NO 96214	DRAWING NO 948929-9901
SCALE	REV. A	SHEET 1 of 15



NOSYMT

LIBRARY DSC2.PXROBJ.PX9OBJ

LIBRARY DSC2.PXROBJ.NEWOBJ

948929-9901 *A PAGE 0002

PHASE 0, PXRATE

INCLUDE (PXRBUF)

INCLUDE (ASM)

INCLUDE (EDT)

INCLUDE (TRAIINT)

INCLUDE (SETREG)

INCLUDE (SETTRA)

INCLUDE (MINIT)

INCLUDE (ERRINT)

INCLUDE (WKSFMG)

INCLUDE (CMSTRP)

INCLUDE (PCOUT)

INCLUDE (GTFELD)

INCLUDE (GTHEX)

INCLUDE (SUPVSR)

INCLUDE (CONVRT)

INCLUDE (ASGLUN)

INCLUDE (CHARIN)

INCLUDE (DISIO)

INCLUDE (DSR733)

INCLUDE (SB)

INCLUDE (RANGE)

INCLUDE (IM)

INCLUDE (IC)

INCLUDE (INWKSP)

INCLUDE (INSPSS)

INCLUDE (MODME)

INCLUDE (MODCR)

INCLUDE (MODREG)

INCLUDE (MODWP)

INCLUDE (PRTMMC)

INCLUDE (PRTSSN)

INCLUDE (PRLDDR)

INCLUDE (OVELAY)

INCLUDE (PROMLD)

INCLUDE (USRPGM)

INCLUDE (RUNPGN)

INCLUDE (SSNAP)

INCLUDE (SCNTBL)

INCLUDE (SIEPR)

INCLUDE (BKPPR)

INCLUDE (CLEAR)

INCLUDE (FIND)

INCLUDE (HXARTH)

INCLUDE (WRTprt)

INCLUDE (FPDEFP)

INCLUDE (TTYPT)

INCLUDE (CMDEFQ)

INCLUDE (PXDATA)

PHASE 1, PXRLLAL

INCLUDE (LALN)

PHASE 1, PXRABS

INCLUDE (ABSOMP)

INCLUDE (ABSLD)

PHASE 1, PXRPRM

INCLUDE (PRMPPG)

PHASE 1, PXRBNPF

INCLUDE (DMBNPF)

PHASE 1, PXRHILO

INCLUDE (DMHL)

END

*END OF BUILD PHASE.



MODULE NAME	LENGTH	PROGRAM ORIGIN	DATE	TIME	MODULE SOURCE
PXRLAL	06B0	4F0C	02/20/77	15:56:47	SDSMAC

DEFINITIONS

NAME	VALUE	NAME	VALUE	NAME	VALUE	NAME	VALUE
*BEG LAL	4F0C	*END LAL	55BA	LALCSR	4FCA		



MODULE NAME	LENGTH	PROGRAM ORIGIN	DATE	TIME	MODULE SOURCE
PXRBUF	0002	0000	11/11/76	08:53:32	SDSMAC
ASM	1B7C	0002	19//77	17:44:10	MIRA99
PXRDT	108E	1B7E	25//77	10:06:29	MIRA99
TRACEM	0558	2C0C			
SETREG	00A6	3164			
SETRAC	009E	320A			
MINIT	0032	32A8	00/00/00	02:12:37	SDSMAC
ERRINT	001A	32DA			
WKSPMG	008C	32F4			
CMSTPP	0148	3380	12/02/76	09:13:41	SDSMAC
PCOUT	0010	34C8			
GTFELD	0088	34D8			
GTHEX	0080	3560			
SUPVSR	009E	35E0			
CONVRT	011E	367E			
ASGLUN	003C	379C			
CHARIN	00D0	37D8			
DISIO	01A6	38A8	00/00/00	20:52:35	SDSMAC
DSR733	0454	3A4E			
SB	0058	3EA2			
RANGE	0046	3EFA			
IM	0012	3F40			
IC	001C	3F52			
INWKSP	0058	3F6E			
INSPSS	0024	3FC6			
MODME	0030	3FEA			
MODCR	0082	401A			
MODREG	0082	409C			
MODWP	005C	411E			
PRTMMC	0096	417A			
PRTSSN	006A	4210			
PRLDDR	017A	427A	09/29/76	11:15:32	SDSMAC
QVELAY	007E	43F4	11/08/76	08:43:11	SDSMAC
PROMLD	002A	4472	00/00/00	17:52:14	SDSMAC
USRPGM	000A	449C			
RUNPGN	0090	44A6	02/20/77	15:42:15	SDSMAC
SSNAP	00B0	4536			
SCNTBL	0030	45E6			
SIEPR	005E	4616			
BKPPR	0064	4674			
CLEAR	003A	46D8			
FIND	00A6	4712			
HXARTH	006A	47B8			
WRTPRT	004A	4822			
FPDEFF	0000	486C	00/00/00	09:09:52	SDSMAC
TTYPT	02A4	486C	11/11/76	08:57:02	SDSMAC
CMDEFO	00E0	4B10	00/00/00	00:15:47	SDSMAC
PXDATA	031C	4BF0			

DEFINITIONS

NAME	VALUE	NAME	VALUE	NAME	VALUE	NAME	VALUE
ACL	330A	ASLUN	379C	ASMSCR	044A	*BDA	000A*
BGNPXR	0000	*BHA	000C*	BKPT	4D5C	BKPTPR	4674
BKSTR	4C08	BKSTRB	4CE0	BLANK	4CA4	BPTAB	4D62
BS	4CAC	BSFLG	4D96	CBD	4BFE	CBDSTG	4C00



NAME	VALUE	NAME	VALUE	NAME	VALUE	NAME	VALUE
CBH	4BF6	CBH0	4BF8	*CBH2	4BFA	CBP	46D8
CDNDT	38E6	*CHB	4BF0	*CHBVAL	4BF2	CLDT	3928
CLEARP	4822	CLST	367E	*CMDMSK	393E	CMDTBL	4B10
CNVRT	368E	COMMA	4CA7	CPLSAV	4D92	CR	4CAA
CRLF	4CB0	CRR	46E4	CRUOFF	1FE0*	CRUPRT	1FA0*
CS	0001*	CSBACK	0000*	CSFWD	0000*	CSOPEN	3A90
CSP	3380	CSRASC	3A98	CSRWND	0000*	CSS	46DE
CSUNLD	0000*	CSWASC	3B5C	CSWEOF	3B0C	*DAB	000B*
DC7331	3A4E	DCTTYT	488E	DFBIAS	4CA2	DMPSPR	4CD4
DUM	007F	EDTCRS	1D24	EFMASK	393D	EI	32DA
ENDBUF	0002	ENTSIZ	000A*	EOR	4CAA	EOSIGN	4CD2
ERBUF	4CB0	ERROR	3858	ERSTR	4CB8	ESC	4CAD
EXCT	449C	FBT	4736	*FPSTRT	FE00*	FPWP	F800*
FREMEM	4DCC	FWD	4712	GETBUF	32FA	GETCHR	37D8
GETFLD	34DC	GETFLN	34D8	GETHEX	3564	GETHXN	3560
HAB	0000	HILIM	4DA2	HXAR	47B8	ICP	3F52
IMP	3F40	INCHAR	4C12	INIMEM	32F6	INIT	32A8
INSCNT	4D98	IO	394E	IRP	40AA	ISRFLG	4D2C
ISS	3FC6	IWP	3F6E	LDABS	4284	LDADDR	4C2A
LDBUF	4C34	LDCBA	4C30	LDCBBC	4C32	LDCC	4C2E
*LDENTY	4C86	LDFLG	4C28	*LDLPT	4C88	LDLUN	4C27
LDNMCC	4C8A	LDOPCD	4C26	*LDPGNM	4C8B	LDPRB	4C24
LDPRT	482C	LDTBL	4C86	LDUFL	427A	*LF	4CAB
LGRASC	3BF8	LGWASC	3D12	LOAD	427E	LOADOV	429A
LOG	0000	LOWLIM	4DA0	*LMP	3302	MEMWP	4DAC
MINUS	4CA6	MODCRU	401C	MODMEM	3FEA	MODWSP	411E
MONCHR	3D86	MRP	40B2	*NOCELS	000A*	NOCMDS	4BEE
NUMBPS	0004*	NUMSNP	0004*	NUMTR	0004*	NUMTTE	0004*
OPEN	0000	*OVLRET	4428	OVLY	43F4	PCOUT	34C8
PERIOD	4CA9	PL	4472	PLUS	4CA8	PRBBUF	4C1A
PRBCC	4C1E	PRBUER	4C18	PRCLOS	49A0	PRCRLF	37F8
PRDASC	49A6	PRDIRT	4A22	PRINT	3800	PRNTC	37FC
PRNTHB	381A	PRNTHN	3820	PRNTHX	3826	PROMPT	4CB4
PROPEN	4994	PRTCRU	418C	PRTMEM	417A	PRTPRD	4CD6
PRTSS	4210	PTCEOF	48B6	PTCLOS	48AE	PTDIFF	4CCA
PTDIRT	4924	PTOPEN	4890	PTOPRW	48A2	*PTP	0002*
PTR	0003	PTRILL	4986	PTSUM	4CC4	PTWASC	48CE
PTWEOF	495C	RANGEX	3F02	*RDASC	0009*	RETBUF	32FE
RGSTR	4CE2	RGSTRB	4CE4	ROMLDR	FFFA*	RR	330E
RUBOUT	4CAE	RUN	44A6	*RWP	3306	SAVWP	4D90
SBP	3EA4	SCTB	45E6	SETLUN	3910	SIE	4616
SIEINT	4646	SIEST	4D8C	SIWEP	4D8A	SIINPG	4D8E
SNAPS	4D36	SNPENT	0008*	SNPTAB	4D3C	SR	3164
SREGN	4BAA	SSS	4536	SSSTR	4CE6	SSSTRB	4CED
STRACE	3226	STRPRT	4D94	*SVC	007C*	SVCALT	4C20
SVCPRB	4C06	SVCSPG	4C0A	SVC5R	3602	SVC5RA	35FE
*SVCUSR	4C0B	*SVCW10	4DA4	*SVCW7	4DA4	SVCWP	4D96
SVCWRT	4C14	SYMT	0001	TERM	44A2	TERMC	4CAF
TRACE	4D26	TRACER	2D02	TRFLGS	0000*	TRHIGH	0004*
TRLNTH	4D2A	TRLQW	0002*	TRNARA	4F0C	TRTBL	4CEE
TRTYPE	0006*	TRV1	0008*	TTTBL	4D2E	*UEFMSK	393F
UEMASK	393C	USRPC	4D9C	USRST	4D9E	USRWP	4D9A
USRWSP	4D80	*WTASC	000B*	ZERO	4CA5		

O

O

O

MODULE NAME	LENGTH	PROGRAM ORIGIN	DATE	TIME	MODULE SOURCE
ABSDMP	0242	4F0C			
ABSLD	0214	514E			

DEFINITIONS

NAME	VALUE	NAME	VALUE	NAME	VALUE	NAME	VALUE
*ABS BUT	514E	ABSLDR	5164	*DPRG	4F7E		

MODULE NAME	LENGTH	PROGRAM ORIGIN	DATE	TIME	MODULE SOURCE
PROMPG	0690	4F0C			

D E F I N I T I O N S

NAME	VALUE	NAME	VALUE	NAME	VALUE	NAME	VALUE
PGBIAS	4F5C	*PFC SI	5020	PROGSZ	4F5E	*PSCSI	546A

MODULE NAME	LENGTH	PROGRAM ORIGIN	DATE	TIME	MODULE SOURCE
DMBNPF	0382	4F0C			

DEFINITIONS

NAME	VALUE	NAME	VALUE	NAME	VALUE	NAME	VALUE
*DMBNPF	4F9C						

MODULE NAME	LENGTH	PROGRAM ORIGIN	DATE	TIME	MODULE SOURCE
DMHL	02D6	4F0C			

D E F I N I T I O N S

NAME	VALUE	NAME	VALUE	NAME	VALUE	NAME	VALUE
*DMHL	4F7C						

>38AA BYTES OF LIST SPACE USED OUT OF >4000

**** LINKING COMPLETED



PRINT ITEM NUMBER	QUANTITY PER ASSEMBLY	UNIT OF ISSUE	DWG. SIZE	PART NUMBER	DESCRIPTION	VENDOR PART NUMBER
0001	REF	EA		0948929-2201	CDS,PXRATE LINK CONTROL-PXR990	
0002	REF	EA		0948928-9901	AL,PXRBUF,RES MCNITOR BUFFER DEF-PXR990	
0002A					PXRATE-ROOT	
0002B					MUST BE FIRST IN PXRATE-	
0002C					ROOT	
0003	REF	EA		0948925-9901	AL,PXRASM,ONE-PASS ASSEMBLER-PXR990	
0003A					PXRATE-ROOT	
0004	REF	EA		0948927-9901	AL,PXREDT,TEXT EDITOR-PXR990	
0004A					PXRATE-ROOT	
0005	REF	EA		0945387-9901	AL, TRACEMOD,TRACE INTERPRETER-PX990	
0005A					PXRATE-ROOT	
0006	REF	EA		0945385-9901	AL, SETREG,SET TRACE REGION-PX990	
0006A					PXRATE-ROOT	
0007	REF	EA		0945386-9901	AL, SETTRACE,SET TRACE FORMAT-PX990	
0007A					PXRATE-ROOT	
0008	REF	EA		0948938-9901	AL,MINIT,INITIALIZE MONITOR-PXR990	
0008A					PXRATE-ROOT	
0009	REF	EA		0945354-9901	AL, ERRINT,ERROR INTERRUPT PROC-PX990	
0009A					PXRATE-ROOT	
0010	REF	EA		0945382-9901	AL, WKSPMG,WORKSPACE MANAGEMENT-PX990	

DRAFTSMAN	DATE	CKD. DRAFTSMAN	DATE	DESIGN ENGINEER	DATE	TITLE
		<i>Larry R. Arants</i>	<i>4-27-77</i>			L4L,PXRATE,CO-RES PROTO SYS MON-PXR990
APPD.-MFG.	DATE	APPD. PROJECT ENGINEER	DATE	RELEASED	DATE	PROJECT NO.
				<i>Larry R. Arants</i>	<i>5/4/77</i>	<i>7506</i>
					PART NUMBER	REV
					LM0948929-9901	A



TEXAS INSTRUMENTS
INCORPORATED

DATE 04/26/77

LIST OF MATERIAL

PAGE 11 of

PART NUMBER	REV
LM0948929-9901	A

PRINT ITEM NUMBER	QUANTITY PER ASSEMBLY	UNIT OF ISSUE	DWG. SIZE	PART NUMBER	DESCRIPTION	VENDOR PART NUMBER
0010A					PXRATE-ROOT	
0011	REF	EA		0945330-9901	AL, CMSTPP,CMD STRING PROC,PROTO-PX990	
0011A					PXRATE-ROOT	
0012	REF	EA		0945369-9901	AL, PCOUT,OUTPUT DATA,FRONT PANEL-PX990	
0012A					PXRATE-ROOT	
0013	REF	EA		0945356-9901	AL, GTFELD,INPUT CHARACTER STRING-PX990	
0013A					PXRATE-ROOT	
0014	REF	EA		0945357-9901	AL, GTHEx,INPUT HEXADECIMAL NUMBER-PX990	
0014A					PXRATE-ROOT	
0015	REF	EA		0945381-9901	AL, SUPVSR,SUPERVISOR CALL I/F-PX990	
0015A					PXRATE-ROOT	
0016	REF	EA		0945352-9901	AL, CONVRT,ASCII/BINARY CONVERSION-PX990	
0016A					PXRATE-ROOT	
0017	REF	EA		0945346-9901	AL, ASGLUN,ASSIGN LUNO-PX990	
0017A					PXRATE-ROOT	
0018	REF	EA		0945348-9901	AL, CHARIN,CHARACTER INPUT-PX990	
0018A					PXRATE-ROOT	
0019	REF	EA		0948939-9901	AL,MPXIOS,I/O DISPATCHER-PXR990	
0019A					PXRATE-ROOT	
0020	REF	EA		0945353-9901	AL, DSR733,733ASR DEVICE DRIVER-PX990	

DRAFTSMAN	DATE	CKD. DRAFTSMAN	DATE	DESIGN ENGINEER	DATE	TITLE				
						LML,PXRATE,CO-RES PROTO SYS MON-PXR990				
APPD.-MFG.	DATE	APPD. PROJECT ENGINEER	DATE	RELEASED	DATE	PROJECT NO.				
						<table border="1"> <tr> <td>PART NUMBER</td> <td>REV</td> </tr> <tr> <td>LM0948929-9901</td> <td>A</td> </tr> </table>	PART NUMBER	REV	LM0948929-9901	A
PART NUMBER	REV									
LM0948929-9901	A									



TEXAS INSTRUMENTS
INCORPORATED

DATE 04/26/77

LIST OF MATERIAL

PAGE 12 of

PART NUMBER	REV
LM0948929-9901	A

PRINT ITEM NUMBER	QUANTITY PER ASSEMBLY	UNIT OF ISSUE	DWG. SIZE	PART NUMBER	DESCRIPTION	VENDOR PART NUMBER
0020A					PXRATE-ROOT	
0021	REF	EA		0945377-9901	AL, SB, SET BREAKPOINT-PX990	
0021A					PXRATE-ROOT	
0022	REF	EA		0945375-9901	AL, RANGE, EXTRACT RANGE/PARAMETERS-PX990	
0022A					PXRATE-ROOT	
0023	REF	EA		0945360-9901	AL, IM, INSPECT MEMORY-PX990	
0023A					PXRATE-ROOT	
0024	REF	EA		0945359-9901	AL, IC, INSPECT CRU PROCESSOR-PX990	
0024A					PXRATE-ROOT	
0025	REF	EA		0945363-9901	AL, INWKSP, INSPECT WORKSPACE-PX990	
0025A					PXRATE-ROOT	
0026	REF	EA		0945362-9901	AL, INSPSS, INSPECT SNAPSHOT-PX990	
0026A					PXRATE-ROOT	
0027	REF	EA		0945365-9901	AL, MODME, MODIFY MEMORY-PX990	
0027A					PXRATE-ROOT	
0028	REF	EA		0945364-9901	AL, MODCR, MODIFY CRU-PX990	
0028A					PXRATE-ROOT	
0029	REF	EA		0945366-9901	AL, MODREG, MODIFY REGISTERS-PX990	
0029A					PXRATE-ROOT	
0030	REF	EA		0945367-9901	AL, MODWP, MODIFY WORKSPACE-PX990	

DRAFTSMAN	DATE	CKD. DRAFTSMAN	DATE	DESIGN ENGINEER	DATE	TITLE
						L 1L, PXRATE, CO-RES PROTO SYS MON-PXR990
APPD.-MFG.	DATE	APPD. PROJECT ENGINEER	DATE	RELEASED	DATE	PROJECT NO.
						PART NUMBER
						LM0948929-9901
						REV
						A



PRINT ITEM NUMBER	QUANTITY PER ASSEMBLY	UNIT OF ISSUE	DWG. SIZE	PART NUMBER	DESCRIPTION	VENDOR PART NUMBER
0030A					PXRATE-ROOT	
0031	REF	EA		0945370-9901	AL, PRTMMC, PRINT MEMORY, CRU RANGES-PX990	
0031A					PXRATE-ROOT	
0032	REF	EA		0945371-9901	AL, PRTSSN, PRINT SNAPSHOT-PX990	
0032A					PXRATE-ROOT	
0033	REF	EA		0948934-9901	AL, PRLDDR, LOADER DRIVER-PXR990	
0033A					PXRATE-ROOT	
0034	REF	EA		0948935-9901	AL, OVELAY, OVERLAY SUPERVISOR-PXR990	
0034A					PXRATE-ROOT	
0035	REF	EA		0948937-9901	AL, PROMLD, PROM LOADER-PXR990	
0035A					PXRATE-ROOT	
0036	REF	EA		0945383-9901	AL, USRPGM, USER PROG SPVSR CALLS-PX990	
0036A					PXRATE-ROOT	
0037	REF	EA		0936187-9901	AL, PUNPGN RUN USER PROGRAM-PXR990	
0037A					PXRATE-ROOT	
0038	REF	EA		0945380-9901	AL, SSNAP, SET SNAPSHOT-PX990	
0038A					PXRATE-ROOT	
0039	REF	EA		0945378-9901	AL, SCNTBL, SCAN TABLES-PX990	
0039A					PXRATE-ROOT	
0040	REF	EA		0945379-9901	AL, SIEPR, SIE DRIVER-PX990	

DRAFTSMAN	DATE	CKD. DRAFTSMAN	DATE	DESIGN ENGINEER	DATE	TITLE
						LML, PXRATE, CO-RES PROTO SYS MON-PXR990
APPD.-MFG.	DATE	APPD. PROJECT ENGINEER	DATE	RELEASED	DATE	PROJECT NO.
						PART NUMBER
						LM0948929-9901
						REV
						A



TEXAS INSTRUMENTS
INCORPORATED

DATE 04/26/77

LIST OF MATERIAL

PAGE 14 of

PART NUMBER	REV
LM0948929-9901	A

PRINT ITEM NUMBER	QUANTITY PER ASSEMBLY	UNIT OF ISSUE	DWG. SIZE	PART NUMBER	DESCRIPTION	VENDOR PART NUMBER
0040A					PXRATE-ROOT	
0041	REF	EA		0945347-9901	AL, BKPPR, BREADPOINT PROCESSOR-PX990	
0041A					PXRATE-ROOT	
0042	REF	EA		0945349-9901	AL, CLEAR, CLEAR COMMAND PROCESSOR-PX990	
0042A					PXRATE-ROOT	
0043	REF	EA		0945355-9901	AL, FIND, SCAN MEMORY FOR VALUES-PX990	
0043A					PXRATE-ROOT	
0044	REF	EA		0945358-9901	AL, HXARTH, HEXADECIMAL ARITHMETIC-PX990	
0044A					PXRATE-ROOT	
0045	REF	EA		0943524-9901	AL, WRTPRT, WRITE PROTECT-PX990	
0045A					PXRATE-ROOT	
0046	REF	EA		0943523-9901	AL, FPDEF, FRONT PANEL DEF'S, PROTO-PX990	
0046A					PXRATE-ROOT	
0047	REF	EA		0948941-9901	AL, TTYPT, PAPER TAPE DSR-PXR990	
0047A					PXRATE-ROOT	
0048	REF	EA		0948930-9901	AL, CMDEF, COMMAND DEFINITION TBL-PXR990	
0048A					PXRATE-ROOT	
0048B					MUST BE NEXT-TO-LAST IN	
0048C					PXRATE-ROOT	
0049	REF	EA		0945372-9901	AL, PXDATA, READ/WRITE DATA AREA-PX990	

DRAFTSMAN	DATE	CKD. DRAFTSMAN	DATE	DESIGN ENGINEER	DATE	TITLE	LML, PXRATE, CO-RES PROTO SYS MON-PXR990	
APPD.-MFG.	DATE	APPD. PROJECT ENGINEER	DATE	RELEASED	DATE	PROJECT NO.	PART NUMBER	REV
							LM0948929-9901	A



PRINT ITEM NUMBER	QUANTITY PER ASSEMBLY	UNIT OF ISSUE	DWG. SIZE	PART NUMBER	DESCRIPTION	VENDOR PART NUMBER
0049A					PXRATE-ROOT	
0049B					MUST BE LAST IN PXRATE-ROOT	
0050	REF	EA		0948926-9901	AL, PXRLAL, LINKING LOADER-PXR990	
0050A					PXRATE-PXRLAL OVERLAY	
0051	REF	EA		0945388-9901	AL, ABSDMP, DUMP ABSOLUTE-PX990	
0051A					PXRATE-PXRABS OVERLAY	
0052	REF	EA		0945389-9901	AL, ABSLD, ABSOLUTE LOADER-PX990	
0052A					PXRATE-PXRABS OVERLAY	
0053	REF	EA		0945391-9901	AL, PROMPG, PROM PROGRAMMER, PART 1-PX990	
0053A					PXRATE-PXRPRO OVERLAY	
0054	REF	EA		0945397-9901	AL, DMBNPF, DUMP BNPf-PX990	
0054A					PXRATE-PXRBNP OVERLAY	
0055	REF	EA		0945325-9901	AL, DMHL, DUMP HI-LO, PROTO-PX990	
0055A					PXRATE-PXRHIL OVERLAY	

DRAFTSMAN	DATE	CKD. DRAFTSMAN	DATE	DESIGN ENGINEER	DATE	TITLE
						L4L, PXRATE, CO-RES PROTO SYS MON-PXR990
APPD.-MFG.	DATE	APPD. PROJECT ENGINEER	DATE	RELEASED	DATE	PROJECT NO.
						PART NUMBER
						LM0948929-9901
						REV
						A

A

APPLICATION		REVISIONS			
NEXT ASSY	USED ON	LTR	DESCRIPTION	DATE	APPROVED
	7506				

NOTE: SOURCE AND OBJECT ON DISC IN SDP 948931.

REV																			
SHEET	1A																		
REV STATUS OF SHEETS	REV SHEET																		

UNLESS OTHERWISE SPECIFIED
 DIMENSIONS ARE IN INCHES
 TOLERANCES
 ANGLES ± 1°
 3 PLACE DECIMALS ± 0.10
 2 PLACE DECIMALS ± 0.2

IDENTIFYING NUMBERS
 SHOWN IN PARENTHESES
 FOR REFERENCE ONLY

INTERPRET DWG IN
 ACCORDANCE WITH
 MIL STD 100

DWN D. TURNER 3-5-77 DATE

CHK

ENGR Bruce Barber 3-11-77

Mamote 3-10-77

APVD 3/14/77

CONTR NO

DESIGN ACTIVITY RELEASE

Larry R. Arments 3-14-77



TEXAS INSTRUMENTS
 INCORPORATED
 Equipment Group Dallas, Texas

AL, PXRASM, ONE-PASS ASSEMBLER-PXR 990

SIZE A CODE IDENT NO 96214

DRAWING NO 948925-9901

SCALE REV

SHEET 1 of 149

0002 *
0003 * TITLE: PXRASM
0004 * REVISION: 03/01/74
0005 * ORIGINAL
0006 * ONE-PASS ASSEMBLER
0007 * 03/15/76
0008 * MODIFIED TO RUN WITH PX9MTR
0009 * 06/28/76
0010 * MODIFIED FOR INCLUSION IN MONITOR BOOTSTRAP
0011 * ALONG WITH TEXT EDITOR
0012 * COMPUTER: 990, 990 MIRA ASSEMBLY
0013 * ABSTRACT: PXRASM IS A ONE-PASS ASSEMBLER WHICH
0014 * OPERATES IN 4K MEMORY WORDS. PXRASM
0015 * ACCEPTS SOURCE INPUT FROM THE ASR733
0016 * CASSETTE. GENERATES AN OBJECT FILE
0017 * ON ASR733 CASSETTE, PRINTS AN ASSEMBLY
0018 * LISTING AND ERROR MESSAGES
0019 * THE ASR733 PRINTER.
0020 * IN 4K MEMORY CONFIGURATION, A
0021 * MINIMUM OF 135 SYMBOLS ARE AVAILABLE WITHOUT
0022 * PREDEFINED REGISTERS, 125 SYMBOLS WITH
0023 * PREDEFINED REGISTERS. EACH ADDITIONAL
0024 * 4K MEMORY CONFIGURED GIVES APPROXIMATELY
0025 * 820 SYMBOLS.
0026 *

```
0029          IDT 'PXRASM'
0030          *
0031          * TITLE:      DATB
0032          *           PXRASM DATA BASE
0033          * REVISION:  03/01/74
0034          *           ORIGINAL
0035          *           03/15/76
0036          *           MODIFIED TO RUN WITH PXRMTB
0037          * COMPUTER:  990, 990 ASSEMBLY
0038          * ABSTRACT:  THE DATA BASE CONTAINS THE SYMBOL
0039          *           AND FORWARD REFERENCE TABLES,
0040          *           I/O BUFFERS, MESSAGES, CHARACTER
0041          *           CLASS TABLE, AND THE OPERAND
0042          *           MNEUMONIC AND DIRECTIVE TABLE.
0043          * CALLING SEQUENCE:
0044          *           NON CALLABLE
0045          *
0046          *
0047          * MAINW REGISTERS
0048          *
0049          0000  NWVAL  EQU  0
0050          0001  GETC   EQU  1
0051          0002  CHAR   EQU  2
0052          0003  RELABS EQU  3
0053          0004  PC     EQU  4
0054          0005  ZCK    EQU  5
0055          0006  SNAM   EQU  6
0056          0007  NWSYM  EQU  7
0057          0008  SPTR   EQU  8
0058          0009  NULL   EQU  9
0059          000A  SFLAGS EQU 10
0060          000A  DBRN   EQU 10
0061          000A  IOPARM EQU 10
0062          000C  TYPE   EQU 12
0063          000D  SYMNPT EQU 13
0064          000E  PRNPT  EQU 14
0065          000F  GVLPT  EQU 15
0066          *
0067          * LPWP REGISTERS
0068          *
0069          0000  ERRG   EQU  0
0070          0001  BLNKK  EQU  1
0071          0002  MSGPT  EQU  3
0072          0004  LMSGPT EQU  4
0073          0006  SCNR   EQU  6
0074          0007  BINDPT EQU  7
0075          0008  EROFST EQU  8
0076          000B  NUMERG EQU 11
0077          000C  PGP    EQU 12
```

```

0079      *
0080      *   CONVWS REGISTERS
0081      *
0082      0000  BINVAL  EQU  0
0083      0001  VALSV   EQU  1
0084      0002  MADD    EQU  2
0085      0003  SCCODE  EQU  3
0086      0007  BHCODE  EQU  7
0087      0008  BDCODE  EQU  8
0088      0009  HCNT    EQU  9
0089      000A  PCNT    EQU 10
0090      000E  CHOPC   EQU 14
0091      *
0092      *   SCWP REGISTERS
0093      *
0094      0000  QTCLS   EQU  0
0095      0001  CLSTYP  EQU  1
0096      0003  RESLT   EQU  3
0097      0004  ERR4    EQU  4
0098      0005  SCPT    EQU  5
0099      0007  NXCPT  EQU  7
0100      0008  INCSC   EQU  8
0101      0009  SCNT   EQU  9
0102      000A  TQC    EQU 10
0103      000B  RTN    EQU 11
0104      000C  SAVE   EQU 12
0105      *
0106      *   SYMBOL REGISTERS
0107      *
0108      0000  SYMPTR  EQU  0
0109      0001  SYMNO   EQU  1
0110      0001  COUNT  EQU  1
0111      0002  FLAGS  EQU  2
0112      0003  SYMNM  EQU  3
0113      0004  NSYM   EQU  4
0114      0005  VALUE  EQU  5
0115      0006  NWSCT  EQU  6
0116      0007  SCNT2  EQU  7
0117      0008  NSBIT  EQU  8
0118      0009  GSHOLD EQU  9
0119      000A  GSHLD2 EQU 10
0120      000C  OUTS   EQU 12
0121      *
0122      *   EXWP REGISTERS
0123      *
0124      0000  RSVL    EQU  0
0125      0004  RESULT2 EQU  4
0126      0005  OPRES   EQU  5
0127      0006  RELVL   EQU  6
0128      0007  RLVAL  EQU  7
0129      0008  OP      EQU  8
0130      0009  SP      EQU  9
0131      000A  JVAL    EQU 10
0132      000A  FL      EQU 10

```

0133	000C	ZCK2	EQU	12
0134	000D	NFND	EQU	13
0135	000E	FSTIM	EQU	14
0136	000F	NGFL	EQU	15
0137		*		
0138		*	LKWP	REGISTERS
0139		*		
0140	0000	SCPTPT	EQU	0
0141	0001	LCKRG	EQU	1
0142	0005	SYM	EQU	5
0143	0006	TABCHR	EQU	6
0144	000A	BLNKR	EQU	10
0145		*		
0146		*	FRWP	REGISTERS
0147		*		
0148	0000	FWDSN	EQU	0
0149	0000	FRTCNT	EQU	0
0150	0001	FRTPT	EQU	1
0151	0001	FRE	EQU	1
0152	0002	NFRT	EQU	2
0153	0003	MLBPT	EQU	3
0154	0004	NFA	EQU	4
0155	0005	PPCRG	EQU	5
0156	0006	FR2	EQU	6
0157	0007	OPPTR	EQU	7
0158	0008	OPWRD	EQU	8
0159	0009	OPCODE	EQU	9
0160	000B	BRANCH	EQU	11
0161		*		
0162		*	OPWP	REGISTERS
0163		*		
0164	0000	EXPREG	EQU	0
0165	0001	CGET	EQU	1
0166	0002	BVAL	EQU	2
0167	0002	TCHR	EQU	2
0168	0003	NXTWD	EQU	3
0169	0003	NXWD	EQU	3
0170	0004	STEP	EQU	4
0171	0004	INSFLG	EQU	4
0172	0005	FLGREG	EQU	5
0173	0006	TEMPIN	EQU	6
0174	0007	INSTRG	EQU	7
0175	0008	THOLD	EQU	8
0176	0008	RSV3	EQU	8
0177	0009	RSV	EQU	9
0178	000A	SOVAL	EQU	10
0179	000C	REGREG	EQU	12
0180	000D	RAPT	EQU	13
0181	000E	EXPRT	EQU	14
0182	000F	COMMA	EQU	15
0183		*		
0184		*	PGWP	REGISTERS
0185		*		
0186	0000	DECPT	EQU	0
0187	0001	LNCTR	EQU	1

0188	0002	PGCNT	EQU	2
0189	0003	TITPT	EQU	3
0190	0004	BINVP	EQU	4
0191		*		
0192		*	FXFWP	REGISTERS
0193		*		
0194	0000	TEMP	EQU	0
0195	0001	PTR1	EQU	1
0196	0002	SABS	EQU	2
0197	0002	PTR2	EQU	2
0198	0003	FRTVAL	EQU	3
0199	0004	RELREG	EQU	4
0200	0006	HPTR1	EQU	6
0201	0007	HPTR1V	EQU	7
0202	0008	PPCPT2	EQU	8
0203	0009	FRTEND	EQU	9
0204	000A	TEMP2	EQU	10
0205	000C	SVAL	EQU	12
0206		*		
0207		*	OTWP	REGISTERS
0208		*		
0209	0000	NOONS	EQU	0
0210	0001	CKSUM	EQU	1
0211	0002	SVN	EQU	2
0212	0003	ABVL	EQU	3
0213	0004	OBJPT	EQU	4
0214	0005	CDALIM	EQU	5
0215	0006	FVL	EQU	6
0216	0007	INSPT	EQU	7
0217	0008	ABDYL	EQU	8
0218	0009	CDLIM	EQU	9
0219	000C	CKPT	EQU	12
0220		*		
0221		*	OTBWP	REGISTERS
0222		*		
0223	0000	INSPT2	EQU	0
0224	0001	ABET	EQU	1
0225	0002	AODD	EQU	2
0226	0003	SBYT	EQU	3
0227	0004	FBYT	EQU	4
0228	0005	ZZERO	EQU	5
0229	0006	SHLD	EQU	6
0230	0007	OBJPT2	EQU	7
0231	0008	BYTFL	EQU	8
0232	0009	OTOBPT	EQU	9
0233	000A	BLNNK	EQU	10
0234	000B	LOCP2	EQU	11
0235	000C	LOCP3	EQU	12

```

0237      *
0238      * LNREG REGISTERS
0239      *
0240      0000 BINVLP EQU 0
0241      0001 PRBF EQU 1
0242      0002 RDBF EQU 2
0243      0003 PRTPOS EQU 3
0244      0004 BLPT EQU 4
0245      0005 RMK EQU 5
0246      0006 LSTPT EQU 6
0247      0007 FST EQU 7
0248      0008 TEMPRG EQU 8
0249      000B PGHC EQU 11
0250      000C RCD1 EQU 12
0251      *
0252      * I/O WORKSPACE EQUATES
0253      *
0254      0000 RDCODP EQU 0
0255      0001 WRCODP EQU 1
0256      0002 IOC EQU 2
0257      0003 IOPLUN EQU 3
0258      0004 FLGS EQU 4
0259      0005 BUFADR EQU 5
0260      0006 BUFLN EQU 6
0261      0007 CHRCNT EQU 7
0262      000A LEN EQU 10
0263      *
0264      * COMMON REGISTER EQUATES
0265      *
0266      0000 R0 EQU 0
0267      0001 R1 EQU 1
0268      0002 R2 EQU 2
0269      0003 R3 EQU 3
0270      0004 R4 EQU 4
0271      0005 R5 EQU 5
0272      0006 R6 EQU 6
0273      0007 R7 EQU 7
0274      0008 R8 EQU 8
0275      0009 R9 EQU 9
0276      000A R10 EQU 10
0277      000B R11 EQU 11
0278      000C R12 EQU 12
0279      000D R13 EQU 13
0280      000E R14 EQU 14
0281      000F R15 EQU 15

```

```

0283 0000          RORG
0284          *
0285          *      DATA
0286          *
0287          *      AORG >20          (REMOVED 07/76)
0288          *
0289 0000      41  ASM      BYTE 'A'
0290 0001      54  TERM     BYTE 'T'
0291 0002      3A  COLON    BYTE ':', >0D      END OF MODULE RECORD
      0003      0D
0292 0004      09  TAB      BYTE >09          TAB CHARACTER
0293 0005      0C  FF       BYTE >0C
0294          REF  SYMT
0295 0006  0000  SYMTBE DATA SYMT          SYMBOL TABLE PTR TO NXT ENTRY
0296          *
0297          *  ERROR MESSAGES
0298          *
0299 0008      20  ERMSG    TEXT ' ** ERR '
0300 0011      00  ERNUM    BYTE 0
0301 0012      20  TEXT     TEXT ' - STMT '
0302 001A      ERLOC     BSS 4
0303 001E      0A  LERADD   BYTE >0A, >0D
      001F      0D
0304 0020      4C  LERMSG   TEXT 'LAST ERR - STMT '      LAST ERR - STMT XXXX
0305 0030      LERLOC     BSS 4
0306 0034      0A  BYTE     BYTE >0A, >0D
      0035      0D

0307          *
0308          *  UNDEFINED SYMBOL ERROR MESSAGES
0309          *
0310 0036      0A  UNDMMSG  BYTE >0A
0311 0037      55  TEXT     TEXT 'UNDEF SMBL'
0312 0041      0A  BYTE     BYTE >0A, >0D
      0042      0D

0313          *
0314          *  ERROR COUNT MESSAGE
0315          *
0316 0043      0A  NUMMSG   BYTE >0A
0317 0044      NUMLOC     BSS 4          NUMBER OF ERRORS
0318 0048      20  TEXT     TEXT ' ERRORS'
0319 004F      0A  BYTE     BYTE >0A, >0D
      0050      0D
0320 0051      20  UNLMSG   TEXT ' LOC '
0321 0056      UNLLOC     BSS 4
0322 005A      0A  BYTE     BYTE >0A, >0D
      005B      0D

0323          *
0324          *  PERIPHERAL DEVICE DEFAULT LOGICAL UNIT NUMBERS
0325          *
0326          *      RORG
0327          *
0328 005C  0000  KBLUND    DATA 0          LOG (KEYBOARD/PRINTER)
0329 005E  0006  PRLUND    DATA 6          LOG (PRINTER)
0330 0060  0007  CLLUND    DATA 7          CS1 (LEFT CASSETTE - INPUT)

```

```
0331 0062 0008 C2LUND DATA 8 CS2 (RIGHT CASSETTE - OUTPUT)
0332 *
0333 * NON I/O MONITOR SUPV CALL BLOCKS
0334 *
0335 0064 0400 ENDPRG DATA >0400 END OF PROGRAM
0336 *
0337 * DEFINE SUPV CALL XOP
0338 *
0339 DXOP SVC.15
0340 *
0341 * MONITOR SUPV CALL CODES FOR I/O
0342 *
0343 0066 00 OPNCOD BYTE >00 OPEN
0344 0067 00 EOFCOD BYTE >00 END OF FILE
0345 0900 RDCOD EQU >0900 READ ASCII
0346 0B00 WRCOD EQU >0B00 WRITE ASCII
0347 *
0348 *
0349 * DATA
0350 * OBJECT RECORD BUFFER
0351 *
0352 0000 WORD1 EQU 0
0353 0002 WORD2 EQU 2
0354 0068 0000 LABPTR DATA 0
0355 006A 0800 EDF DATA >800 EXT DEF MARKER
0356 006C 0000 NEWFRF DATA 0
0357 006E 01 Z1 BYTE 1
0358 006F 10 ROAB BYTE >10
0359 0070 0008 EIGHT DATA 8
0360 0072 000A TENS DATA 10
0361 0074 0064 DATA 100
0362 0076 03E8 DATA 1000
0363 *
0364 0078 OBJRCD BSS 74
0365 *
0366 * TITLE
0367 *
0368 00C2 0C TITLE BYTE >0C
0369 00C3 TITFX BSS 50 TITLE CHARACTERS
0370 00F5 20 TPAGE TEXT / PAGE /
0371 00FB PAGNUM BSS 4 PAGE NUMBER
0372 00FF 0A BYTE >0A
0373 0100 0A00 LFCR DATA >0A00 LINE FEED/CARRIAGE RETURN
0374 0100 LF EQU LFCR
0375 0101 CR EQU LFCR+1
```

```

0377      *
0378      *      TABLE OVERFLOW MESSAGE
0379      *
0380 0102    20 ABORT TEXT  ' *ABORT*'
0381 010A    0A      BYTE >0A, >0D
      010B    0D
0382 010C   0000 ABRTFL DATA 0      ABORT FLAG
0383      *
0384      *      END
0385      *
0386 010E    41 ENDMSG TEXT 'ASM/TERM?'
0387 0118    0D      BYTE >0D
0388      *
0389      *      CASSETTE NOT READY MESSAGE
0390      *
0391 0119    52 RDYMSG TEXT 'RDY OBJ-TYPE CR'
0392 0128    0D      BYTE >0D
0393      *      EVEN
0394      *
0395      *      INSTRUCTION BUFFER
0396      *
0397 012A   2020 IOBUF DATA >2020
0398 012C    04 RCDNUM BSS 4      RECORD NUMBER
0399 0130   2020 DATA >2020
0400 0132    04 LOCNUM BSS 4      LOCATION OF INSTRUCTION
0401 0136   2020 DATA >2020
0402 0138    04 INSVL BSS 4      INSTRUCTION VALUE
0403 013C    01 RELMRK BSS 1      RELOCATABILITY MARK
0404 013D    20      BYTE >20
0405 013E    62 PRBUF BSS 62      PRINT BUFFER
0406 017C    62 INSBUF BSS 62
0407      *      EVEN
0408      *
0409      *      ASSORTED EQUATES
0410      *
0411 0085' OBJST EQU OBJRCD+13
0412 000D EOL EQU >D
0413 0040 ATSGN EQU >40
0414 002A ASTR EQU >2A
0415 002F OPCNT EQU 47
0416 0016 SIXTN EQU >16      SIXTEEN
0417 4000 IOERR EQU >4000      I/O ERROR STATUS
0418 2000 EOF EQU >2000      END OF FILE STATUS
0419      *
0420      *      CHARACTER CLASS
0421      *
0422      *      CLASS CHARACTER VALUE
0423 01BA    00      BYTE 0, 0, 0, 0      OTHER      0
      01BB    00
      01BC    00
      01BD    00
0424 01BE    07      BYTE 7, 0, 0, 8      +      1
      01BF    00
      01C0    00

```

0425	01C1	08			
	01C2	00	BYTE 0, 0, 3, 1	-	2
	01C3	00			
	01C4	03			
	01C5	01			
0426	01C6	00	BYTE 0, 2, 0, 4	*	3
	01C7	02			
	01C8	00			
	01C9	04			
0427	01CA	06	BYTE 6, 6, 6, 6	/	4
	01CB	06			
	01CC	06			
	01CD	06			
0428	01CE	06	BYTE 6, 6, 6, 6	ALPHA	5
	01CF	06			
	01D0	06			
	01D1	06			
0429	01D2	06	BYTE 6, 6, 0, 0	NUM	6
	01D3	06			
	01D4	00			
	01D5	00			
0430	01D6	00	BYTE 0, 0, 9, 0	QUOTE	7
	01D7	00			
	01D8	09			
	01D9	00			
0431	01DA	00	BYTE 0, 5, 5, 5	QUOTE	8
	01DB	05			
	01DC	05			
	01DD	05			
0432	01DE	05	BYTE 5, 5, 5, 5	>	9
	01DF	05			
	01E0	05			
	01E1	05			
0433	01E2	05	BYTE 5, 5, 5, 5		
	01E3	05			
	01E4	05			
	01E5	05			
0434	01E6	05	BYTE 5, 5, 5, 5		
	01E7	05			
	01E8	05			
	01E9	05			
0435	01EA	05	BYTE 5, 5, 5, 5		
	01EB	05			
	01EC	05			
	01ED	05			
0436	01EE	05	BYTE 5, 5, 5, 5		
	01EF	05			
	01F0	05			
	01F1	05			
0437	01F2	05	BYTE 5, 5, 5, 0		
	01F3	05			
	01F4	05			
	01F5	00			
0438	01F6	00	BYTE 0, 0, 0, 0		
	01F7	00			

```
01F8 00
01F9 00
0439 *
0440 * OPERATOR MNEMONIC AND DIRECTIVE TABLE
0441 *
0442 OPTBL
0443 01FA 4D TEXT 'MOV'
0444 01FD 00 BYTE 0, >C0, 1
01FE C0
01FF 01
0445 0200 4D TEXT 'MOVB'
0446 0204 D001 DATA >D001
0447 0206 4C49 DATA 'LI', 0, >0204
0208 0000
020A 0204
0448 020C 41 BYTE 'A', 0, 0, 0, >A0, 1
020D 00
020E 00
020F 00
0210 A0
0211 01
0449 0212 4142 DATA 'AB', 0, >B001
0214 0000
0216 B001
0450 0218 4149 DATA 'AI', 0, >0224
021A 0000
021C 0224
0451 021E 41 TEXT 'ANDI'
0452 0222 0244 DATA >244
0453 0224 424C DATA 'BL', 0, >0682
0226 0000
0228 0682
0454 022A 42 TEXT 'BLWP'
0455 022E 0402 DATA >402
0456 0230 42 TEXT 'BYTE'
0457 0234 040A DATA >40A
0458 0236 44 TEXT 'DATA'
0459 023A 050A DATA >50A
0460 023C 54 TEXT 'TEXT'
0461 0240 110A DATA >110A
0462 0242 43 BYTE 'C', 0, 0, 0, >80, 1
0243 00
0244 00
0245 00
0246 00
0247 01
0463 0248 4342 DATA 'CB', 0, >9001
024A 0000
024C 9001
0464 024E 4349 DATA 'CI', 0, >0284
0250 0000
0252 0284
0465 0254 43 TEXT 'CLR'
0466 0257 00 BYTE 0, 4, >C2
0258 04
```

	0259	02		
0467	025A	44	TEXT	'DEC'
0468	025D	00	BYTE	0, 6, 2
	025E	06		
	025F	02		
0469	0260	44	TEXT	'DECT'
0470	0264	0642	DATA	>642
0471	0266	44	TEXT	'DEF'
0472	0269	00	BYTE	0, 6, >A
	026A	06		
	026B	0A		
0473	026C	52	TEXT	'REF'
0474	026F	00	BYTE	0, >F, >A
	0270	0F		
	0271	0A		
0475	0272	45	TEXT	'EQU'
0476	0275	00	BYTE	0, >A, >A
	0276	0A		
	0277	0A		
0477	0278	49	TEXT	'INC'
0478	027B	00	BYTE	0, 5, >82
	027C	05		
	027D	82		
0479	027E	49	TEXT	'INCT'
0480	0282	0502	DATA	>502
0481	0284	4A	TEXT	'JEQ'
0482	0287	00	BYTE	0, >13, 7
	0288	13		
	0289	07		
0483	028A	4A	TEXT	'JGT'
0484	028D	00	BYTE	0, >15, 7
	028E	15		
	028F	07		
0485	0290	4A48	DATA	'JH', 0, >1B07
	0292	0000		
	0294	1B07		
0486	0296	4A	TEXT	'JHE'
0487	0299	00	BYTE	0, >14, 7
	029A	14		
	029B	07		
0488	029C	4A4C	DATA	'JL', 0, >1A07
	029E	0000		
	02A0	1A07		
0489	02A2	4A	TEXT	'JLE'
0490	02A5	00	BYTE	0, >12, 7
	02A6	12		
	02A7	07		
0491	02A8	4A	TEXT	'JLT'
0492	02AB	00	BYTE	0, >11, 7
	02AC	11		
	02AD	07		
0493	02AE	4A	TEXT	'JMP'
0494	02B1	00	BYTE	0, >10, 7
	02B2	10		
	02B3	07		

0495	02B4	4A	TEXT	'JNC'
0496	02B7	00	BYTE	0, >17, 7
	02B8	17		
	02B9	07		
0497	02BA	4A	TEXT	'JNE'
0498	02BD	00	BYTE	0, >16, 7
	02BE	16		
	02BF	07		
0499	02C0	4A	TEXT	'JND'
0500	02C3	00	BYTE	0, >19, 7
	02C4	19		
	02C5	07		
0501	02C6	4A	TEXT	'JOC'
0502	02C9	00	BYTE	0, >18, 7
	02CA	18		
	02CB	07		
0503	02CC	4A	TEXT	'JOP'
0504	02CF	00	BYTE	0, >1C, 7
	02D0	1C		
	02D1	07		
0505	02D2	5254	DATA	'RT', 0, >045B
	02D4	0000		
	02D6	045B		
0506	02D8	52	TEXT	'RTWP'
0507	02DC	0380	DATA	>380
0508	02DE	53	BYTE	'S', 0, 0, 0, >60, 1
	02DF	00		
	02E0	00		
	02E1	00		
	02E2	60		
	02E3	01		
0509	02E4	5342	DATA	'SB', 0, >7001
	02E6	0000		
	02E8	7001		
0510	02EA	44	TEXT	'DIV'
0511	02ED	00	BYTE	0, >3C, 6
	02EE	3C		
	02EF	06		
0512	02F0	4D	TEXT	'MPY'
0513	02F3	00	BYTE	0, >38, 6
	02F4	38		
	02F5	06		
0514	02F6	53	TEXT	'SLA'
0515	02F9	00	BYTE	0, >A, 9
	02FA	0A		
	02FB	09		
0516	02FC	53	TEXT	'SRA'
0517	02FF	00	BYTE	0, 8, 9
	0300	08		
	0301	09		
0518	0302	53	TEXT	'SRC'
0519	0305	00	BYTE	0, >B, 9
	0306	0B		
	0307	09		
0520	0308	53	TEXT	'SRL'

0521	0306	00	BYTE 0, 9, 9
	030C	09	
	030D	09	
0522	030E	53	TEXT 'SWPB'
0523	0312	06C2	DATA >6C2
0524	0314	53	TEXT 'STCR'
0525	0318	3406	DATA >3406
0526	031A	4C	TEXT 'LDCR'
0527	031E	3006	DATA >3006
0528	0320	53	TEXT 'SB0'
0529	0323	00	BYTE 0, >1D, 3
	0324	1D	
	0325	03	
0530	0326	53	TEXT 'STST'
0531	032A	02C5	DATA >2C5
0532	032C	53	TEXT 'STWP'
0533	0330	02A5	DATA >2A5
0534	0332	54	TEXT 'TITL'
0535	0336	120A	DATA >120A
0536	0338	58	TEXT 'XOP'
0537	033B	00	BYTE 0, >2C, 6
	033C	2C	
	033D	06	
0538	033E	58	TEXT 'XOR'
0539	0341	00	BYTE 0, >28, 6
	0342	28	
	0343	06	
0540	0344	53	TEXT 'SZC'
0541	0347	00	BYTE 0, >40, 1
	0348	40	
	0349	01	
0542	034A	53	TEXT 'SZCB'
0543	034E	5001	DATA >5001
0544	0350	5442	DATA 'TB', 0, >1F03
	0352	0000	
	0354	1F03	
0545	0356	53	TEXT 'SOC'
0546	0359	00	BYTE 0, >E0, 1
	035A	E0	
	035B	01	
0547	035C	53	TEXT 'SOCB'
0548	0360	F001	DATA >F001
0549	0362	53	TEXT 'SBZ'
0550	0365	00	BYTE 0, >1E, 3
	0366	1E	
	0367	03	
0551	0368	53	TEXT 'SET0'
0552	036C	0702	DATA >702
0553	036E	52	TEXT 'RORG'
0554	0372	100A	DATA >100A
0555	0374	52	TEXT 'RSET'
0556	0378	0360	DATA >360
0557	037A	4F	TEXT 'ORI'
0558	037D	00	BYTE 0, 2, >64
	037E	02	

	037F	64		
0559	0380	50	TEXT	'PAGE'
0560	0384	0E0A	DATA	>E0A
0561	0386	4E	TEXT	'NEG'
0562	0389	00	BYTE	0, 5, 2
	038A	05		
	038B	02		
0563	038C	4E	TEXT	'NOP'
0564	038F	00	BYTE	0
0565	0390	1000	DATA	>1000
0566	0392	4C	TEXT	'LWPI'
0567	0396	02E8	DATA	>2E8
0568	0398	4C	TEXT	'LIMI'
0569	039C	0308	DATA	>308
0570	039E	4C	TEXT	'LIST'
0571	03A2	0D0A	DATA	>D0A
0572	03A4	41	TEXT	'ABS'
0573	03A7	00	BYTE	0, 7, >42
	03A8	07		
	03A9	42		
0574	03AA	41	TEXT	'AORG'
0575	03AE	010A	DATA	>10A
0576	03B0	42	BYTE	'B', 0, 0, 0, 4, >42
	03B1	00		
	03B2	00		
	03B3	00		
	03B4	04		
	03B5	42		
0577	03B6	42	TEXT	'BES'
0578	03B9	00	BYTE	0, 2, >A
	03BA	02		
	03BB	0A		
0579	03BC	42	TEXT	'BSS'
0580	03BF	00	BYTE	0, 3, >A
	03C0	03		
	03C1	0A		
0581	03C2	43	TEXT	'CKOF'
0582	03C6	03C0	DATA	>3C0
0583	03C8	43	TEXT	'CKON'
0584	03CC	03A0	DATA	>3A0
0585	03CE	43	TEXT	'COC'
0586	03D1	06	BYTE	0, >20, 6
	03D2	20		
	03D3	06		
0587	03D4	43	TEXT	'CZC'
0588	03D7	00	BYTE	0, >24, 6
	03D8	24		
	03D9	06		
0589	03DA	44	TEXT	'DORG'
0590	03DE	070A	DATA	>70A
0591	03E0	55	TEXT	'UNL'
0592	03E3	00	BYTE	0, >13, >A
	03E4	13		
	03E5	0A		
0593	03E6	58	BYTE	'X', 0, 0, 0, 4, >82

ERFX

NOP AND ERROR INSTRUCTION FILL

	03E7	00	
	03E8	00	
	03E9	00	
	03EA	04	
	03EB	82	
0594	03EC	44	TEXT 'DXOP'
0595	03F0	0B0A	DATA >B0A
0596	03F2	45	TEXT 'END'
0597	03F5	00	BYTE 0, 9, >A
	03F6	09	
	03F7	0A	
0598	03F8	45	TEXT 'EVEN'
0599	03FC	0B0A	DATA >B0A
0600	03FE	49	TEXT 'IDLE'
0601	0402	0340	DATA >340
0602	0404	49	TEXT 'IDT'
0603	0407	00	BYTE 0, >C, >A
	0408	0C	
	0409	0A	
0604	040A	49	TEXT 'INV'
0605	040D	00	BYTE 0, 5, >42
	040E	05	
	040F	42	
0606	0410	4C	TEXT 'LREX'
0607	0414	03E0	DATA >3E0
0608	0416	4C	TEXT 'LDD'
0609	0419	00	BYTE 0
0610	041A	07C2	DATA >07C2
0611	041C	4C	TEXT 'LDS'
0612	041F	00	BYTE 0
0613	0420	0782	DATA >0782
0614			OPTEND
0615	0422	4C	TEXT 'LMF'
0616	0425	00	BYTE 0
0617	0426	0329	DATA >0329

```
0620 *
0621 * WORKSPACE AREA FOR MAIN DRIVER
0622 *
0623 MAINW
0624 0428 0000 NWVL DATA 0 R0 NWVAL
0625 042A 116E' DATA GETCHR R1 GETC
0626 042C 0000 DATA 0 R2 CHAR
0627 042E 0000 DATA 0 R3 RELABS
0628 0430 0000 PCVL DATA 0 R4 PC
0629 0432 0000 DATA 0 R5 ZCK
0630 0434 0000 DATA 0 R6 SNAM
0631 0436 0000 NSMBL DATA 0 R7 NWSYM
0632 0438 0000 DATA 0 R8 SPTR
0633 043A 0000 DATA 0 R9 NULL & I/O RTN STATUS
0634 043C 0000 DATA 0 R10 SFLAGS, DBRN, IOPARM
0635 043E 0000 DATA 0 R11
0636 0440 0000 DATA 0 R12 TYPE
0637 0442 0D38' DATA SYMBOL+SYMMN+SYMMN R13 SYMNPT
0638 0444 186A' DATA PRNTLN R14 PRNPT
0639 0446 0D6E' DATA GETSVL R15 GVLPT
```

```

0641 * TITLE: START/PXRASM
0642 * PXRASM MAIN DRIVER
0643 * REVISION: 03/01/74
0644 * ORIGINAL
0645 * 03/15/76
0646 * MODIFIED TO RUN WITH PXRMR
0647 * COMPUTER: 990, ASSEMBLY
0648 * ABSTRACT: START BRANCHES TO INITIALIZATION ROUTINE
0649 * FOR HEADING, MEMORY SIZE, AND PREDEFINED
0650 * REGISTERS FIRST TIME PXRASM EXECUTED AFTER
0651 * LOAD. SUBSEQUENT EXECUTIONS BRANCH TO PXRASM.
0652 * PXRASM CONTAINS THE CODE FOR INITIALIZATION
0653 * OF FLAGS AND BUFFERS, AND THE MAIN DRIVER
0654 * WHICH IS RESPONSIBLE FOR THE GENERAL
0655 * SCAN OF AN INPUT LINE (PROCESSING OF
0656 * LABELS AND EXAMINATION OF OPERATOR
0657 * FIELD), AND THE BASIC CALLS TO IDENTI-
0658 * FICATION AND PROCESSING ROUTINES.
0659 * CALLING SEQUENCE:
0660 * NON-CALLABLE
0661 * ADDITIONAL NOTES:
0662 * WORKSPACE = MAINW(SHARED WITH DIPR)
0663 * SHARES MAINW AS A WORKSPACE. PXRASM PERFORMS
0664 * ALL NECESSARY INITIALIZATION TO KEEP PXRASM
0665 * REENRANT.
0666 * ROUTINES CALLED: PRINT, PRINTN, KEYIN, OPEN,
0667 * SYMCLR, RDRCD, LKCHAR, PRNTLN, SCAN, SRCSYM,
0668 * GETSCT, GETSVL, GETCHR, ER2, OPSRCH, AEVEN,
0669 * OPTYPE, XOPG, ER6, AEVEN
0670 DEF ASMCSR
0671 ASMCSR
0672 START
0673 0448 02E0 LWPI MAINW INIT WORKSPACE POINTER
0674 044A 0428'
0674 044C 06A0 BL @INIT INITIALIZE PROGRAM
0674 044E 1A6C'
0675 PXRASM
0676 0450 02E0 LWPI MAINW
0676 0452 0428'
0677 0454 0300 LIMI 2 INIT STATUS
0677 0456 0002
0678 0458 020A LI IOPARM, LFCR
0678 045A 0100'
0679 045C 0420 BLWP @PRINT
0679 045E 19CE'
0680 ENDACT
0681 0460 020A LI IOPARM, ENDMSG PRINT END MESSAGE
0681 0462 010E'
0682 0464 0420 BLWP @PRINTN
0682 0466 19D6'
0683 0468 020A LI IOPARM, INSBUF INPUT RESPONSE
0683 046A 017C'
0684 046C 0420 BLWP @KEYIN
0684 046E 198A'

```

```

0685 0470 9820      CB   @INSBUF,@TERM      IS IT TERMINATE?
      0472 017C✓
      0474 0001✓
0686 0476 1602      JNE  ENDA                NO
0687 0478 2FE0      SVC  @ENDPRG             END OF PROGRAM SUPV CALL
      047A 0064✓

0688                                ENDA
0689 047C 9820      CB   @INSBUF,@ASM       IS IT ASSEMBLE?
      047E 017C✓
      0480 0000✓
0690 0482 16EE      JNE  ENDACT             NO
0691 0484 C2A0      MOV  @C1LUND,IOPARM    OPEN CASSETTE 1
      0486 0060✓
0692 0488 0420      BLWP @OPEN
      048A 1964✓
0693 048C C2A0      MOV  @C2LUND,IOPARM    OPEN CASSETTE 2
      048E 0062✓
0694 0490 0420      BLWP @OPEN
      0492 1964✓

0695                                *
0696                                *   INITIALIZATION OF FLAGS, BUFFERS, AND
0697                                *   BUFFER POINTERS
0698                                *
0699                                *   BATCH
0700 0494 05A0      INC  @LIST              SET LIST FLAG
      0496 1856✓
0701 0498 05A0      INC  @OTWP+NCONS+NCONS SET NON-CONSECUTIVE FLAG
      049A 16E5✓
0702 049C 04E0      CLR  @NUMERR           CLEAR ERROR COUNT
      049E 0B44✓
0703 04A0 04E0      CLR  @LNCNT           CLEAR LINE COUNT
      04A2 1590✓
0704 04A4 04E0      CLR  @NPAGE           CLEAR PAGE NUMBER
      04A6 1592✓
0705 04A8 04E0      CLR  @DORGFL          CLEAR DORG ENCOUNTERED FLAG
      04AA 11C8✓
0706 04AC 04E0      CLR  @ABRTFL          CLEAR ABORT FLAG
      04AE 010C✓
0707 04B0 C820      MOV  @LFCR,@LERADD    TRUNCATE LAST ERR MSG
      04B2 0100✓
      04B4 001E✓
0708 04B6 0208      LI   SPTR,OBJRCD      INITIALIZE OBJECT BUFFER
      04B8 0078✓
0709 04BA C020      MOV  @OTBWP+ZZERO+ZZERO,NWVAL
      04BC 17D2✓
0710 04BE CE00      MOV  NWVAL,*SPTR+     ZERO PROGRAM LENGTH
0711 04C0 CE00      MOV  NWVAL,*SPTR+
0712 04C2 DE00      MOV  NWVAL,*SPTR+
0713                                CLRBF
0714 04C4 DE20      MOV  @BLANK,*SPTR+    CLEAN OUT OBJECT BUFFER
      04C6 17DC✓
0715 04C8 0288      CI   SPTR,TPAGE
      04CA 00F5✓
0716 04CC 11FB      JLT  CLRBF
0717 04CE D820      MOV  @FF,@TITLE      STORE FORM FEED

```

```

04D0 0005'
04D2 00C2'
0718 04D4 C820      MOV  @OBJSTP,@OTWP+OBJPT+OBJPT
04D6 0BD6'
04D8 16EE'
0719 04DA 04E0      CLR  @OTBWP+BYTFL+BYTFL CLEAR BYTE FLAG
04DC 17D8'
0720 04DE 04E0      CLR  @NEWFRF          CLEAR NEW FWD REF FLAG
04E0 006C'
0721 04E2 0208      LI   SPTR,FRT-1     INITIALIZE POINTER TO FRT
04E4 009F
0722 04E6 C808      MOV  SPTR,@FRWP+MLBPT+MLBPT
04E8 11B6'
0723 04EA C820      MOV  @SYMTBE,@ENDST MARK END OF SYMBOL TABLE
04EC 0006'
04EE 0B38'
0724 04F0 04C4      CLR  PC             INITIALIZE PROGRAM COUNTER AN
0725 04F2 C804      MOV  PC,@RELPC
04F4 0C18'
0726 04F6 0203      LI   RELABS,1      RELOCATABILITY VALUE
04F8 0001
0727 04FA 04E0      CLR  @RCDCNT       CLEAR RECORD COUNT
04FC 1862'
0728 04FE 0208      LI   SPTR,FRT      CLEAR SYMBOL TABLE
0500 00A0
0729                                CLRS
0730 0502 04F8      CLR  *SPTR+
0731 0504 8808      C    SPTR,@ENDST
0506 0B38'
0732 0508 11FC      JLT  CLRS
0733                                AREAD
0734                                *
0735                                * READ AND PROCESS A RECORD
0736                                *
0737 050A 02E0      LWPI MAINW
050C 0428'
0738 050E C160      MOV  @NEWFRF,ZCK   HAVE ANY FWD REFS BEEN RES'D
0510 006C'
0739 0512 1305      JEQ  AR2           NO
0740 0514 C160      MOV  @OTBWP+BYTFL+BYTFL,ZCK CONVENIENT TO RESOLVE?
0516 17D8'
0741 0518 1602      JNE  AR2           NO
0742 051A 0420      BLWP @SYMCLR       CLEAR SYMBOL FOR RESOLVE
051C 0DF6'
0743                                AR2
0744 051E 05A0      INC  @FSTPT        SET FIRST-PRINT-OF-LINE FLAG
0520 1858'
0745 0522 04E0      CLR  @SCWP+INCSC+INCSC CLEAR INSIDE-STRING FLAG
0524 0C1C'
0746 0526 04E0      CLR  @LABSN        CLEAR LABEL SEEN FLAG
0528 15A6'
0747 052A C804      MOV  PC,@PPC       INITIALIZE PRESENT PC
052C 11BA'
0748 052E C820      MOV  @LNREG+RDBF+RDBF,@SCWP+SCPT+SCPT
0530 184E'

```


0749	0532	0C16			
	0534	0420	BLWP	@RDRCD	READ A RECORD
	0536	190A			
0750	0538	0420	BLWP	@LKCHAR	LOOK AT CHARACTER
	053A	115C			
0751	053C	0282	CI	CHAR, '*'	IS CHARACTER AN ASTERISK
	053E	002A			
0752	0540	1604	JNE	APROC	NO, PROCESS LINE
0753				APRLN	
0754				APERLN	
0755	0542	02E0	LWPI	MAINW	REESTABLISH WORKSPACE PTR
	0544	0428			
0756	0546	041E	BLWP	*PRNPT	PRINT LINE
0757	0548	10E0	JMP	AREAD	READ NEXT LINE
0758				APROC	
0759			*		
0760			*	PROCESS LABEL FIELD	
0761			*		
0762	054A	0282	CI	CHAR, >20	IS CHARACTER A BLANK
	054C	0020			
0763	054E	133A	JEQ	PROOF	YES, NO LABEL
0764	0550	0420	BLWP	@SCAN	NO, GET LABEL
	0552	0C2C			
0765	0554	1000	NOP		RETURN FROM SCAN
0766	0556	0420	BLWP	@SRCSYM	SEARCH SYMBOL TABLE
	0558	0D94			
0767	055A	1021	JMP	ANTFND	SYMBOL NOT IN TABLE - JUMP
0768	055C	C2A0	MOV	@SYMBOL+FLAGS+FLAGS, SFLAGS	IN TABLE
	055E	0D36			
0769	0560	028A	CI	SFLAGS, 8	IS SYMBOL ALREADY DEFINED
	0562	0008			
0770	0564	1503	JGT	ANODEF	NO
0771				AMDFER	
0772	0566	0420	BLWP	@ERS	MULTIPLY DEFINED SYMBOL
	0568	0B5A			
0773	056A	102C	JMP	PROOF	PROCESS OPERATOR
0774				ANODEF	
0775	056C	028A	CI	SFLAGS, 11	IS SYMBOL EXTERNAL REF
	056E	0008			
0776	0570	15FA	JGT	AMDFER	YES, AMBIGUOUS DEF
0777	0572	0200	LI	NWVAL, 4	NO, MARK AS DEFINED
	0574	0004			
0778	0576	028A	CI	SFLAGS, 10	IS SYMBOL PREVIOUSLY REF'D
	0578	000A			
0779	057A	1104	JLT	AREF	NO, CONTINUE
0780	057C	0220	RI	NWVAL, -4	YES, MARK AS RECENTLY DEFINED
	057E	FFFC			
0781	0580	05A0	INC	@NEWFRF	SET NEW FWD REF FLAG
	0582	006C			
0782				AREF	
0783	0584	028A	CI	SFLAGS, 11	IS SYMBOL EXT DEF'D
	0586	000B			
0784	0588	1601	JNE	ADEF	YES
0785	058A	0580	INC	NWVAL	NO, ADD 1 TO FLAG VALUE
0786				ADEF	

0787	058C	C0C3	MOV	RELABS, RELABS	IS SYMBOL CODE RELOCATABLE
0788	058E	1601	JNE	AREL	YES
0789	0590	05C0	INCT	NWVAL	NO, ADD TWO TO FLAG VALUE
0790			AREL		
0791	0592	0A30	SLA	NWVAL, 3	POSITION FLAGS
0792	0594	0420	BLWP	@GETSCT	GET SYMBOL COUNT
	0596	0D5C			
0793	0598	E020	SOC	@SYMBOL+COUNT+COUNT, NWVAL	
	059A	0D34			
0794	059C	1008	JMP	ANF2	PLACE SYMBOL INTO TABLE
0795			ANTFND		
0796	059E	C020	MOV	@SYMBOL+NWSCT+NWSCT, NWVAL	
	05A0	0D3E			
0797	05A2	0220	AI	NWVAL, ->30	MARK AS DEF'D RELOCATABLE
	05A4	FFD0			
0798	05A6	C0C3	MOV	RELABS, RELABS	IS CODE PRESENTLY RELOCATABLE
0799	05A8	1602	JNE	ANF2	YES
0800	05AA	0220	AI	NWVAL, >10	NO, MARK AS ABSOLUTE
	05AC	0010			
0801			ANF2		
0802	05AE	C220	MOV	@SYMBOL+SYMPTR+SYMPTR, SPTR	
	05B0	0D32			
0803	05B2	06C0	SWPB	NWVAL	PLACE NEW FLAG IN ENTRY
0804	05B4	D600	MOVB	NWVAL, *SPTR	
0805	05B6	041F	BLWP	*GVLPT	POINT TO SYMBOL VALUE
0806	05B8	C19D	MOV	*SYMPTR, SNAM	
0807	05BA	C584	MOV	PC, *SNAM	LET PC BE PRESENT VALUE
0808	05BC	C808	MOV	SPTR, @LABPTR	SAVE LABEL ENTRY
	05BE	0068			
0809	05C0	05A0	INC	@LABSN	SET LABEL SEEN FLAG
	05C2	15A6			
0810			*		END OF LABEL PROCESSING
0811			PROOP		
0812			*		
0813			*	PROCESS OPERATOR FIELD	
0814			*		
0815	05C4	0411	BLWP	*GETC	GET CHARACTER
0816	05C6	0282	CI	CHAR, >20	IS SYMBOL A BLANK
	05C8	0020			
0817	05CA	1303	JEQ	ANXCHR	YES, GET NEXT CHARACTER
0818	05CC	0420	BLWP	@ER2	SYNTAX ERROR
	05CE	0B4E			
0819	05D0	10B8	JMP	APERLN	PRINT ERROR LINE
0820			ANXCHR		
0821	05D2	0420	BLWP	@LKCHAR	LOOK AT CHARACTER
	05D4	115C			
0822	05D6	0282	CI	CHAR, EOL	END OF LINE?
	05D8	0000			
0823	05DA	13B3	JEQ	APERLN	YES, LABEL-ONLY CARD OR BLANK
0824	05DC	0420	BLWP	@SCAN	NO, GET OPERATOR
	05DE	0C2C			
0825	05E0	102A	JMP	OPERX	NON-SYMBOL IN OPERATOR FLD
0826	05E2	C820	MOV	@SCWP+RESLT+RESLT, @FRWP+OPWRD+OPWRD	
	05E4	0C12			
	05E6	11C8			

0827	05E8	C160	MOV	@SYMBL+4,ZCK	IS SYMBOL <= 4 CHARACTERS
	05EA	114E'			
0828	05EC	161A	JNE	NOOPF	NO - SYMBOL NOT IN OP TABLE
0829	05EE	0420	BLWP	@OPSRCH	SEARCH FOR OPERATOR
	05F0	0EEA'			
0830	05F2	1017	JMP	NOOPF	OPERATOR NOT IN TABLE
0831	05F4	0411	BLWP	*GETC	GET CHARACTER
0832	05F6	0282	CI	CHAR.>20	IS THE CHARACTER A BLANK
	05F8	0020			
0833	05FA	1307	JEQ	CLROP	YES, CONTINUE
0834					
0835	05FC	0420	BLWP	@ER2	SYNTAX ERROR
	05FE	0B4E'			
0836					
0837	0600	C820	MOV	@ERFX,@FRWP+OPCODE+OPCODE	SET NOP
	0602	0390'			
	0604	1102'			
0838	0606	04E0	CLR	@FRWP+BRANCH+BRANCH	SET ZERO BRANCH
	0608	1106'			
0839					
0840	060A	8820	C	@FRWP+BRANCH+BRANCH,@TENS	DIRECTIVE?
	060C	1106'			
	060E	0072'			
0841	0610	1304	JEQ	NOADJ	YES, DO NOT FIND WORD BNDRY
0842	0612	06A0	BL	@AEVEN	GET WORD BOUNDARY
	0614	0B10'			
0843	0616	C804	MOV	PC,@PPC	UPDATE PRESENT PC
	0618	11BA'			
0844					
0845	061A	0420	BLWP	@OPTYE	PROCESS OPERATOR
	061C	127C'			
0846					
0847	061E	0460	B	@AREAD	READ A RECORD
	0620	050A'			
0848					
0849	0622	0420	BLWP	@SRCSYM	SEARCH SYMBOL TABLE
	0624	0D94'			
0850	0626	1007	JMP	OPERX	NOT IN TABLE / ERROR
0851	0628	8820	C	@SYMBOL+FLAGS+FLAGS,@EIGHT	IS THIS A DXOP
	062A	0D36'			
	062C	0070'			
0852	062E	1603	JNE	OPERX	NO - ERROR
0853	0630	0420	BLWP	@XOPG	YES, PROCESS DEFINED OP
	0632	149A'			
0854	0634	10F4	JMP	NXCRD	READ NEXT CARD
0855	0636	0420	OPERX	BLWP @ER6	UNRECOGNIZABLE OPERATOR
	0638	0B5E'			
0856	063A	10E2	JMP	NOPST	SET UP NOP

```

0859 * TITLE: DIFR
0860 * DIRECTIVE PROCESSORS
0861 * REVISION: 03/01/74
0862 * ORIGINAL
0863 * 03/15/76
0864 * MODIFIED TO RUN WITH PXRMTN
0865 * COMPUTER: 990, ASSEMBLY
0866 * ABSTRACT:
0867 * ALL DIRECTIVE ENTRIES IN THE OPERATOR MNEMONIC
0868 * AND DIRECTIVE TABLE CONTAIN THE BRANCH 10 AND
0869 * AN OPCODE ENTRY EQUIVALENT TO THEIR DIRECTIVE
0870 * NUMBERS. THIS IS USED TO BRANCH TO THE PROPER
0871 * DIRECTIVE PROCESSOR. RETURN FROM ANY PROCESSOR
0872 * IS EITHER TO 'AREAD' OR 'APRLN' IN DRIV99 TO
0873 * READ A NEW RECORD OR PRINT A LINE RESPECTIVELY.
0874 * CALLING SEQUENCE:
0875 * FROM OPTYPE ONLY. THE CALL IS PART OF THE
0876 * OPERATOR CLASS BRANCH TABLE FOR WHICH THE
0877 * TENTH ENTRY IS DATA DIFR
0878 * ADDITIONAL NOTES:
0879 * WORKSPACE = MAINW (SHARED WITH START/PXRASM)
0880 * ROUTINES CALLED: WDE, GETSVL, PCPRN, EXEVL,
0881 * TRUNK, BINHEX, OUTBYT, PRNTLN, NXTFLD, REVEN, EXPR,
0882 * OUTOBJ, SCAN, SRCSYM, ER5, ER2, GETCHR, SYMCLR,
0883 * CLROBJ, OUTSYM, REFDEF, LKCHAR, WRITE, BINDEC,
0884 * PRINT, GETSCT, ER4, PGHDR, OUTNW2, EXPR
0885 DIFR
0886 063C 02E0 LMPI MAINW
0886 063E 0428'
0887 *
0888 * BRANCH TO APPROPRIATE PROCESSOR
0889 *
0890 0640 D2A0 MOVB @FRWP+OPCODE+OPCODE, DBRN GET BRANCH VALUE
0890 0642 11C2'
0891 0644 097A SRL DBRN, 7 ADJUST FOR JUMP
0892 0646 C2AA MOV @JMPTB8-2(DBRN), DBRN
0892 0648 064A'
0893 064A 045A B *DBRN JUMP TO PROPER PROCESSOR
0894 JMPTB8
0895 064C 0672' DATA AORG 1 AORG
0896 064E 06BA' DATA ABES 2 BES
0897 0650 06D4' DATA ABSS 3 BSS
0898 0652 06E2' DATA ABYTE 4 BYTE
0899 0654 071A' DATA ADATA 5 DATA
0900 0656 0766' DATA DEF 6 DEF
0901 0658 0798' DATA ADORG 7 DORG
0902 065A 07C6' DATA ADXOP 8 DXOP
0903 065C 0806' DATA AEND 9 END
0904 065E 08E0' DATA AEQU 10 EQU
0905 0660 093C' DATA EVEN 11 EVEN
0906 0662 0992' DATA AIDT 12 IDT
0907 0664 09AC' DATA ALIST 13 LIST
0908 0666 09B2' DATA APAGE 14 PAGE
0909 0668 09C0' DATA REF 15 REF
    
```

0910	066A	09F0'	DATA ARORG	16	RORG
0911	066C	0A4C'	DATA ATXT	17	TEXT
0912	066E	0944'	DATA ATITL	18	TITL
0913	0670	0AA8'	DATA AUNL	19	UNL

```

0915      *
0916      * PROCESS 'AORG' COMMAND
0917      *
0918      AAORG
0919      0672 05A0      INC  @OTWP+NCONS+NCONS SET NON-CONSECUTIVE FLAG
           0674 16E6
0920      0676 C0C3      MOV  RELABS,RELABS      IS CODE PRESENTLY RELOCATABLE
0921      0678 1306      JEQ  AAORG2              NO
0922      067A C160      MOV  @DORGFL,ZCK        IS CODE PRESENTLY IN DORG?
           067C 11C8
0923      067E 1602      JNE  AAORG1              YES
0924      0680 C804      MOV  PC,@RELPC          NO, SAVE RELOC VALUE
           0682 0C18
0925      0684 04C3      AAORG1 CLR RELABS        MARK CODE AS ABSOLUTE
0926      0686 04E0      AAORG2 CLR @DORGFL      RESET DORG FLAG
           0688 11C8
0927      068A 06A0      BL   @WDE                GET WELL DEFINED EXPR
           068C 0AC0
0928      068E C160      MOV  @EXWP+RELVL+RELVL,ZCK EXPRESSION RELOCATABLE?
           0690 0F28
0929      0692 1302      JEQ  AAORG3              NO - GOOD
0930      0694 0460      B    @SYNTAX             YES, SYNTAX ERROR
           0696 0790
0931      AAORG3
0932      0698 C120      MOV  @EXWP+RESLT+RESLT,PC MARK NEW LOC
           069A 0F22
0933      069C C160      MOV  @LABSN,ZCK          LABEL PRESENT?
           069E 15A6
0934      06A0 1309      JEQ  AAORG4              NO
0935      06A2 C1E0      MOV  @LABPTR,NWSYM       YES, POINT TO IT
           06A4 0068
0936      06A6 C807      MOV  NWSYM,@SYMBOL+SYMPTR+SYMPTR
           06A8 0D32
0937      06AA F5E0      SOCB @ROAB,*NWSYM        MARK AS ABSOLUTE
           06AC 006F
0938      06AE 041F      BLWP *GVLPT              POINT TO SYMBOL VALUE
0939      06B0 C190      MOV  *SYMNPT,SNAM        ADJUST SYMBOL VALUE
0940      06B2 C584      MOV  PC,*SNAM
0941      AAORG4
0942      06B4 06A0      BL   @PCPRN              PLACE PC IN OUTPUT LINE
           06B6 1490
0943      06B8 1013      JMP  PRG2                 PRINT OUTPUT LINE
0944      *
0945      * PROCESS 'BES' COMMAND
0946      *
0947      ABES
0948      06BA 05A0      INC  @OTWP+NCONS+NCONS SET NON-CONSECUTIVE FLAG
           06BC 16E6
0949      06BE 06A0      BL   @EXEVL              EVALUATE EXPRESSION
           06C0 0AAE
0950      06C2 C820      MOV  @LABPTR,@SYMBOL+SYMPTR+SYMPTR
           06C4 0068
           06C6 0D32
0951      06C8 041F      BLWP *GVLPT              POINT TO SYMBOL VALUE
    
```

0952	06CA	C21D	MOV	*SYMNPT, SPTR	
0953	06CC	C604	MOV	PC, *SPTR	MODIFY SYMBOL VALUE
0954	06CE	06A0	BL	@PCPRN	PLACE PC IN OUTPUT LINE
	06D0	1490			
0955	06D2	1006	JMP	PRG2	PRINT LINE
0956			*		
0957			*	PROCESS 'BSS' COMMAND	
0958			*		
0959			ABSS		
0960	06D4	05A0	INC	@OTWP+NCONS+NCONS	SET NON-CONSECUTIVE FLAG
	06D6	16E6			
0961	06D8	06A0	BL	@PCPRN	PLACE PC IN OUTPUT LINE
	06DA	1490			
0962	06DC	06A0	BL	@EXEVL	EVALUATE EXPRESSION
	06DE	0AAE			
0963			PRG2		
0964	06E0	1059	JMP	PRG	PRINT OUTPUT LINE
0965			*		
0966			*	PROCESS 'BYTE' COMMAND	
0967			*		
0968			ABYTE		
0969	06E2	C804	MOV	PC, @PPC	ADJUST PRESENT PC
	06E4	11BA			
0970	06E6	06A0	BL	@WDE	GET WELL DEFINED EXPRESSION
	06E8	0AC0			
0971	06EA	C020	MOV	@EXWP+RESLT+RESLT, NWVAL	
	06EC	0F22			
0972	06EE	D000	MOVW	NWVAL, NWVAL	TEST RESULT FOR LIMITS
0973	06F0	1302	JEQ	ABYTE2	NO TRUNCATION
0974	06F2	06A0	BL	@TRUNK	TEST TRUNCATION
	06F4	112C			
0975			ABYTE2		
0976	06F6	0240	ANDI	NWVAL, >FF	TRUNCATE
	06F8	00FF			
0977	06FA	0420	BLWP	@BINHEX	CONVERT TO ASCII
	06FC	0BE0			
0978	06FE	0428	DATA	NWVL	LOCATION OF BYTE VALUE
0979	0700	0138	DATA	INSVAL	LOCATION IN OUTPUT LINE
0980	0702	C820	MOV	@BLANK, @INSVAL	CLEAR TOP PART OF WORD
	0704	17DC			
	0706	0138			
0981	0708	06A0	BL	@PCPRN	PLACE PC IN OUTPUT LINE
	070A	1490			
0982	070C	0584	INC	PC	ADJUST PC
0983	070E	0420	BLWP	@OUTBYT	OUTPUT BYTE
	0710	17E8			
0984	0712	041E	BLWP	*PRNPT	PRINT LINE
0985	0714	06A0	BL	@NXTFLD	CHECK FOR NEXT FIELD
	0716	0ACC			
0986	0718	10E4	JMP	ABYTE	GET NEXT BYTE

```

0988      *
0989      *   PROCESS 'DATA' COMMAND
0990      *
0991      *   ADATA
0992      071A 06A0      BL   @AEVEN      GET WORD BOUNDARY
           071C 0B10
0993      *   ADATA2
0994      071E C804      MOV  PC,@PPC      ADJUST PRESENT PC
           0720 11BA
0995      0722 0420      BLWP @EXPR      GET EXPRESSION
           0724 0F42
0996      0726 100C      JMP  ADAT2      NO FWD REF FOUND
0997      *
0998      *   OUTPUT FORWARD REFERENCE
0999      *
1000      0728 06A0      BL   @PCPRN      PLACE PC IN OUTPUT LINE
           072A 1490
1001      072C 05C4      INCT PC      INCREMENT PC
1002      072E 0202      LI   CHAR,>2D2D  PUT IN MINUS SIGNS
           0730 2D2D
1003      0732 C802      MOV  CHAR,@INSVAL  PLACE INTO OUTPUT LINE
           0734 0138
1004      0736 C802      MOV  CHAR,@INSVAL+2
           0738 013A
1005      073A 05A0      INC  @OTWP+NCONS+NCONS  SET NON-CONSECUTIVE FLAG
           073C 16E6
1006      073E 100F      JMP  ADAT4
1007      *   ADATA2
1008      0740 0420      BLWP @BINHEX     CONVERT TO ASCII
           0742 0BE0
1009      0744 0F22      DATA EXWP+RESLT+RESLT  EXPRESSION RESULT
1010      0746 0138      DATA INSVAL     INSTRUCTION LOCATION
1011      0748 06A0      BL   @PCPRN     PLACE PC IN OUTPUT LINE
           074A 1490
1012      074C 05C4      INCT PC      ADJUST PC
1013      074E C160      MOV  @EXWP+RELVL+RELVL,ZCK  EXPRESSION TYPE?
           0750 0F28
1014      0752 1303      JEQ  ADAT3     ABSOLUTE - SKIP
1015      0754 D820      MOVB @QT,@RELMRK  RELOCATABLE - SET APOSTROPHE
           0756 1A6B
           0758 013C
1016      *   ADATA3
1017      075A 0420      BLWP @OUTOBJ     PLACE OBJECT IN BUFFER
           075C 1706
1018      *   ADATA4
1019      075E 041E      BLWP *PRNPT     PRINT LINE
1020      0760 06A0      BL   @NXTFLD    CHECK NEXT FIELD
           0762 0ACC
1021      0764 100C      JMP  ADATA2     MORE DATA - CONTINUE
1022      *
1023      *   PROCESS 'DEF' COMMAND
1024      *
1025      *   DEF
1026      0766 041E      BLWP *PRNPT     PRINT LINE
    
```


1027			DEFF		
1028	0768	0420	BLWP	@SCAN	GET OPERAND
	076A	0C20			
1029	076C	100E	JMP	DFER	NO SYMBOL FOUND
1030	076E	0420	BLWP	@SRCSYM	SEARCH SYMBOL TABLE
	0770	0D94			
1031	0772	1005	JMP	RINSYM	SYMBOL NOT IN TABLE - GOOD
1032	0774	0420	BLWP	@ERS	MULTIPLY DEFINED SYMBOL
	0776	0B5A			
1033			ADEF2		
1034	0778	06A0	BL	@NXTFLD	CHECK NEXT FIELD
	077A	0ACC			
1035	077C	10F5	JMP	DEFF	MORE SYMBOLS - CONTINUE
1036			RINSYM		
1037	077E	C220	MOV	@SYMBOL+SYMPTR+SYMPTR, SPTR	
	0780	0D32			
1038	0782	020C	LI	TYPE, SIXTN	MARK AS UNREF'D EXT DEF
	0784	0010			
1039	0786	660C	S	TYPE, *SPTR	
1040	0788	10F7	JMP	ADEF2	CHECK FOR MORE SYMBOLS
1041			DFER		
1042	078A	0420	BLWP	@ER2	SYNTAX ERROR IN DEF
	078C	0B4E			
1043	078E	10F4	JMP	ADEF2	CHECK NEXT FIELD
1044			SYNTAX		
1045	0790	0420	BLWP	@ER2	SYNTAX ERROR
	0792	0B4E			
1046			PRG		
1047	0794	0460	B	@APRLN	PRINT A LINE
	0796	0542			
1048			*		
1049			*	PROCESS 'DORG' COMMAND	
1050			*		
1051			ADORG		
1052	0798	05A0	INC	@OTWP+NCONS+NCONS	SET NON-CONSECUTIVE FLAG
	079A	16E6			
1053	079C	C160	MOV	@DORGFL, ZCK	IS CODE PRESENTLY IN DORG?
	079E	11C8			
1054	07A0	1606	JNE	ADORG2	YES, CONTINUE
1055	07A2	C0C3	MOV	RELABS, RELABS	NO, IS CODE NOW RELOCATABLE?
1056	07A4	1302	JEQ	ADORG1	NO
1057	07A6	C804	MOV	PC, @RELPC	YES, SAVE RELOCATABLE PC
	07A8	0C18			
1058			ADORG1		
1059	07AA	0720	SETO	@DORGFL	SET DORG ENCOUNTERED FLAG
	07AC	11C8			
1060			ADORG2		
1061	07AE	06A0	BL	@WDE	GET WELL-DEFINED EXPRESSION
	07B0	0ACC			
1062	07B2	C160	MOV	@EXWP+RELVL+RELVL, ZCK	EXPRESSION RELOCATABLE?
	07B4	0F28			
1063	07B6	1603	JNE	RLDORG	YES, RELOCATABLE DORG
1064	07B8	04C3	CLR	RELABS	MARK CODE AS ABSOLUTE
1065	07BA	0460	B	@AORG3	FINISH PROCESSING LIKE AORG
	07BC	0698			

```

1066                                RLDORG
1067 07BE 0203 LI RELABS.1 MARK CODE AS RELOCATABLE
      07C0 0001
1068 07C2 0460 B @ARORG3 FINISH PROCESSING LIKE RORG
      07C4 0A24'

1069 *
1070 * PROCESS 'DXOP' COMMAND
1071 *
1072 ADXOP
1073 07C6 0420 BLWP @SCAN GET LABEL
      07C8 0C2C'
1074 07CA 10E2 JMP SYNTAX NO SYMBOL - ERROR
1075 07CC 0420 BLWP @SRCSYM SEARCH SYMBOL TABLE
      07CE 0D94'
1076 07D0 1003 JMP ADX2 NOT IN TABLE - GOOD
1077 07D2 0420 BLWP @ERS IN TABLE - MULTIBLY DEFINED
      07D4 0B5A'
1078 07D6 10DE JMP PRG PRINT LINE
1079 ADX2
1080 07D8 0320 MOV @SYMBOL+NWSCT+NWSCT,TYPE
      07DA 0D3E'
1081 07DC 022C AI TYPE,->18 MARK AS DXOP
      07DE FFES
1082 07E0 0A9C SLA TYPE,8
1083 07E2 0220 MOV @SYMBOL+SYMPTR+SYMPTR,SPTR
      07E4 0D32'
1084 07E6 060C MOVB TYPE,*SPTR PLACE IN SYMBOL TABLE
1085 07E8 0411 BLWP *GETC GET CHARACTER
1086 07EA 0282 CI CHAR.>2C IS IT A COMMA
      07EC 002C
1087 07EE 16D0 JNE SYNTAX NO SYNTAX ERROR
1088 07F0 02E0 LWPI OPWP ADJUST WORKSPACE
      07F2 1256'
1089 07F4 05A0 BL @RA GET TERM FOR LEVEL INTERRUPT
      07F6 1552'
1090 07F8 02E0 LWPI MAINW REESTABLISH WP
      07FA 0428'
1091 07FC 041F BLWP *GVLPT POINT TO SYMBOL VALUE
1092 07FE 019D MOV *SYMNPT,SNAM
1093 0800 05A0 MOV @OPWP+REGREG+REGREG,*SNAM PLACE LEVEL IN VALUE
      0802 125E'
1094 0804 10C7 JMP PRG PRINT LINE
1095 *
1096 * PROCESS END COMMAND
1097 *
1098 REND
1099 0806 041E BLWP *PRNPT PRINT LINE
1100 0808 0160 MOV @NEWFRF,ZCK CHECK NEW FWD REF FLAG
      080A 006C'
1101 080C 1302 JEQ ENDAL NONE NEW
1102 080E 0420 BLWP @SYMLCR SOME NEW - RESOLVE REFS
      0810 0DF6'

1103 ENDAL
1104 0812 0820 C @OTWP+OBJPT+OBJPT,@OBJBUF
      0814 16EE'
    
```

1105	0816	08D8'			
1106	0818	1302	JEQ	RFDFFX	NO - CONTINUE
	081A	0420	BLWP	@CLROBJ	CLEAR OBJECT BUFFER
	081C	17BE'			
1107				RFDFFX	
1108	081E	0420	BLWP	@OUTSYM	OUTPUT ERROR 1 MESSAGES
	0820	121C'			
1109	0822	0420	BLWP	@REFDEF	OUTPUT REFS. DEFS. UNDEFS.
	0824	0E2E'			
1110	0826	020A	LI	SFLAGS, OBJRCD	POINT INTO OBJECT
	0828	0078'			
1111	082A	0420	BLWP	@LKCHAR	IS THERE AN END VECTOR?
	082C	115C'			
1112	082E	0282	CI	CHAR, >0D	
	0830	000D			
1113	0832	131C	JEQ	NOEV	NO
1114	0834	0420	BLWP	@SCAN	YES, CHECK SYMBOL
	0836	0C2C'			
1115	0838	104E	JMP	EREND	NO SYMBOL - ERROR
1116	083A	0420	BLWP	@SRCSYM	SEARCH SYMBOL TABLE
	083C	0D94'			
1117	083E	104B	JMP	EREND	NOT IN TABLE - ERROR
1118	0840	C1A0	MOV	@SYMBOL+FLAGS+FLAGS, SNAM	CHECK FLAGS
	0842	0D36'			
1119	0844	0286	CI	SNAM, 8	IS SYMBOL DEFINED
	0846	0008			
1120	0848	1446	JHE	EREND	NO - ERROR
1121	084A	C220	MOV	@OTWP+OBJPT+OBJPT, SPTR	YES, POINT TO OBJECT
	084C	16EE'			
1122	084E	DE20	MOVB	@ZZ1, *SPTR+	MARK AS ABSOLUTE ENTRY ADDR
	0850	0B40'			
1123	0852	0246	ANDI	SNAM, 2	TEST FOR ABSOLUTE SYMBOL
	0854	0002			
1124	0856	1603	JNE	ENDV	YES, DO NOT ADJUST
1125	0858	0608	DEC	SPTR	NO, ADJUST FOR RELOCATABLE
1126	085A	BE20	AB	@Z1, *SPTR+	
	085C	006E'			
1127				ENDV	
1128	085E	041F	BLWP	*GVLPT	POINT TO SYMBOL VALUE
1129	0860	0420	BLWP	@BINHEX	CONVERT
	0862	0BE0'			
1130	0864	0D3C'	DATA	SYMBOL+VALUE+VALUE	SYMBOL VALUE TO HEX ASCII
1131	0866	0079'	DATA	OBJRCD+1	& STORE IN OBJECT RECORD
1132	0868	022A	AI	SFLAGS, 5	MODIFY IDT PLACEMENT
	086A	0005			
1133				NOEV	
1134	086C	C0C3	MOV	RELABS, RELABS	CHECK RELOCATABILITY
1135	086E	1602	JNE	RELEND	RELOCATABLE - CONTINUE
1136	0870	C120	MOV	@RELPC, PC	NO - GET LAST RELOC LOCN
	0872	0C18'			
1137				RELEND	
1138	0874	DEA0	MOVB	@OTBWP+ZZERO+ZZERO, *SFLAGS+	
	0876	17D2'			
1139	0878	C80A	MOV	SFLAGS, @EVLOC	INITIALIZE IDT ENTRY
	087A	0886'			

```

1140 087C 0820      MOV  @BLANK,NWVAL      CREATE BLANK
      087E 17DC
1141 0880 0420      BLWP @BINHEX          CONVERT PC
      0882 0BE0
1142 0884 0430      DATA PCVL          LENGTH OF PROGRAM
1143 0886 0886 EVLOC DATA #          LOCATION IN OBJECT
1144 0888 022A      RI   SFLAGS, 4      POINT TO IDT NAME AREA
      088A 0004
1145 088C 0208      LI   SPTR, -8        SET CHARACTER COUNT
      088E FFF8
1146                                ENDLP
1147 0890 DE80      MOVB NWVAL, *SFLAGS+  BLANK OUT IDT NAME
1148 0892 0588      INC  SPTR
1149 0894 11FD      JLT  ENDLP
1150 0896 C80A      MOV  SFLAGS, @OTWP+OBJPT+OBJPT
      0898 16EE
1151 089A 0420      BLWP @CLR0BJ         CLEAR OBJECT
      089C 17BE
1152 089E 020A      LI   IOPARM, COLON   WRITE MODULE SEPARATOR RECORD
      08A0 0002
1153 08A2 0420      BLWP @WRITE
      08A4 19C2
1154 08A6 C820      MOV  @NUMERR, @BINVL CONVERT
      08A8 0B44
      08AA 0BC0
1155 08AC 0420      BLWP @BINDEC         NUMBER OF ERROR
      08AE 0BF4
1156 08B0 0044      DATA NUMLOC         PLACE ERROR COUNT INTO LINE
1157 08B2 020A      LI   IOPARM, NUMMSG
      08B4 0043
1158 08B6 0420      BLWP @PRINT
      08B8 19CE
1159 08BA C160      MOV  @NUMERR, ZCK    IF NO ERRORS SKIP
      08BC 0B44
1160 08BE 1304      JEQ  ENDEXT
1161 08C0 020A      LI   IOPARM, LERMSG
      08C2 0020
1162 08C4 0420      BLWP @PRINT
      08C6 19CE
1163                                ENDEXT
1164 08C8 C160      MOV  @ABRTFL, ZCK    IS ABORT FLAG SET
      08CA 010C
1165 08CC 1602      JNE  ENDRST         YES, RESTART
1166 08CE 0460      B    @BATCH         RESTART ASSEMBLER
      08D0 0494
1167                                ENDRST
1168 08D2 0460      B    @WREOF         WRITE EOF AND RESTART
      08D4 192E
1169 08D6 0420      EREND BLWP @ER2
      08D8 0B4E
1170                                ERNOEV
1171 08DA 020A      LI   SFLAGS, OBJRCD
      08DC 0078
1172 08DE 10C6      JMP  NOEV
1173                                *
    
```

```

1174 * PROCESS 'EQU' COMMAND
1175 *
1176 *   EQU
1177 08E0 06A0 BL @WDE GET WELL DEFINED EXPRESSION
    08E2 0AC0
1178 08E4 C160 MOV @LABSN,ZCK WAS LABEL USED
    08E6 15A6
1179 08E8 1602 JNE EQU2 YES - GOOD
1180 08EA 0460 B @SYNTAX NO. SYNTAX ERROR
    08EC 0790
1181 EQU2
1182 08EE 04C0 CLR NWVAL CLEAR VALUE
1183 08F0 C220 MOV @LABPTR,SPTR POINT TO SYMTAB ENTRY FOR LAB
    08F2 0068
1184 08F4 C808 MOV SPTR,@SYMBOL+SYMPTR+SYMPTR
    08F6 0D32
1185 08F8 D318 MOVB *SPTR,TYPE
1186 08FA 028C CI TYPE,>2000 IS CODE RECENTLY DEFINED
    08FC 2000
1187 08FE 1102 JLT EQU4 YES
1188 0900 0220 AI NWVAL,4 NO. MARK NEW SYMBOL NOT
    0902 0004
1189 EQU4
1190 0904 C160 MOV @EXWP+RELV+RELV,ZCK EXPRESSION RELOCATABLE
    0906 0F28
1191 0908 1602 JNE EQREL YES
1192 090A 05C0 INCT NWVAL NO. MARK SYMBOL ABSOLUTE
1193 090C 1003 JMP EQAB
1194 EQREL
1195 090E D820 MOVB @QT,@RELMRK PUT APOSTROPHE IN LINE
    0910 1A6B
    0912 013C
1196 EQAB
1197 0914 2320 COC @EDF,TYPE EXTERNAL DEF?
    0916 006A
1198 0918 1601 JNE EQDF YES
1199 091A 0580 INC NWVAL NO. ADJUST FLAG
1200 EQDF
1201 091C 0A30 SLA NWVAL,3 ADJUST FLAG IN WORD
1202 091E 0420 BLWP @GETSCT GET SYMBOL COUNT
    0920 0D5C
1203 0922 A020 R @SYMBOL+COUNT+COUNT,NWVAL
    0924 0D34
1204 0926 0A80 SLA NWVAL,8
1205 0928 D600 MOVB NWVAL,*SPTR MOVE IN NEW FLAGS
1206 092A 041F BLWP *GVLPT POINT TO SYMBOL VALUE
1207 092C C190 MOV *SYMNPT,SNAM REPLACE VALUE
1208 092E C5A0 MOV @EXWP+RESLT+RESLT,*SNAM
    0930 0F22
1209 0932 0420 BLWP @BINHEX CONVERT VALUE FOR PRINTING
    0934 0BE0
1210 0936 0F22 DATA EXWP+RESLT+RESLT EXPRESSION VALUE
1211 0938 0138 DATA INSVL LOCATION IN OUTPUT LINE
1212 093A 1002 JMP PRG3
1213 *
    
```

```

1214 * PROCESS 'EVEN' COMMAND
1215 *
1216 EVEN
1217 093C 06A0 BL @AEVEN MAKE PROGRAM COUNTER EVEN
      093E 0B10'
1218 0940 0460 PRG3 B @APRLN
      0942 0542'

1219 *
1220 * PROCESS 'TITL' COMMAND
1221 *
1222 ATITL
1223 0944 020A LI DBRN, 25 PREPARE FOR 50 CHARACTERS
      0946 0019
1224 0948 0207 LI NWSYM, TITFX POINT TO TITLE AREA
      094A 00C3'

1225 TITFX2
1226 094C C00A MOV DBRN, NWVAL SAVE COUNT
1227 094E 0A10 SLA NWVAL, 1
1228 0950 C207 MOV NWSYM, SPTR SAVE POINTER
1229 ZT
1230 0952 DE20 MOVB @BLANK, *SPTR+ CLEAR AREA
      0954 17DC'
1231 0956 0600 DEC NWVAL
1232 0958 15FC JGT ZT
1233 095A 0411 BLWP *GETC GET CHARACTER
1234 095C 0282 CI CHAR, >27 IS CHARACTER A QUOTE
      095E 0027
1235 0960 1673 JNE SYNTAX2 NO - SYNTAX ERROR
1236 0962 05A0 INC @SCWP+INCSC+INCSC SET INSIDE STRING FLAG
      0964 001C'

1237 TIT0
1238 0966 0420 BLWP @SCAN GET TWO CHARACTERS
      0968 002C'
1239 096A 1000 NOP (DUMMY EXIT FROM SCAN)
1240 096C C020 MOV @SCWP+RESLT+RESLT, NWVAL GET WORD OF CHRS
      096E 0012'
1241 0970 1300 JEQ TIT03 IF ZERO - END HIT
1242 0972 DDC0 MOVB NWVAL, *NWSYM+ PLACE INTO STRING
1243 0974 06C0 SWPB NWVAL GET SECOND CHARACTER
1244 0976 D000 MOVB NWVAL, NWVAL IS SECOND CHARACTER ZERO
1245 0978 1309 JEQ TIT02 YES - END AT ONE CHARACTER
1246 097A DDC0 MOVB NWVAL, *NWSYM+ PL&CE INTO STRING
1247 097C 060A DEC DBRN DECREMENT COUNT
1248 097E 15F3 JGT TIT0 CONTINUE IF MORE CHARACTERS
1249 0980 0411 BLWP *GETC GET CHARACTER
1250 0982 0282 CI CHAR, >27 IS CHARACTER A QUOTE
      0984 0027
1251 0986 1302 JEQ TIT03 YES - END NORMALLY
1252 0988 0420 BLWP @ER4 NO - TRUNCATION ERROR
      098A 0B56'

1253 TIT02
1254 TIT03
1255 098C 04E0 CLR @SCWP+INCSC+INCSC CLEAR INSIDE STRING FLAG
      098E 001C'
1256 0990 1015 JMP RRG READ NEXT RECORD
    
```

```

1257      *
1258      *   PROCESS 'IDT' COMMAND
1259      *
1260      AIDT
1261      0992 041E      BLWP *PRNPT      PRINT LINE
1262      0994 8820      C      @OTWP+OBJPT+OBJPT,@OBJSTP POINT INTO OBJECT
           0996 16EE'
           0998 0BD6'
1263      099A 1303      JEQ  IDTFX      RIGHT LOCATION - CONTINUE
1264      099C 0420      BLWP @ER2      TOO LATE - SYNTAX ERROR
           099E 0B4E'
1265      09A0 1000      JMP  RRG      READ NEXT RECORD
1266      IDTFX
1267      09A2 020A      LI   DBRN,4    SET COUNT
           09A4 0004
1268      09A6 0207      LI   NWSYM,OBJRCD+5 POINT TO AREA FOR FILL
           09A8 007D'
1269      09AA 1000      JMP  TITFX2    FILL IN STRING
1270      *
1271      *   PROCESS 'LIST' COMMAND
1272      *
1273      ALIST
1274      09AC 05A0      INC  @LIST     SET LIST FLAG
           09AE 1856'
1275      09B0 1005      JMP  RRG      READ NEXT RECORD
1276      *
1277      *   PROCESS 'PAGE' COMMAND
1278      *
1279      APAGE
1280      09B2 C160      MOV  @LIST,ZCK IS LIST FLAG SET
           09B4 1856'
1281      09B6 1302      JEQ  RRG      NO - READ NEXT RECORD
1282      09B8 0420      BLWP @PGHDR   YES - PRINT PAGE HEADER
           09BA 15AE'
1283      09BC 0460      RRG  B      @AREAD READ NEXT RECORD
           09BE 050A'
1284      *
1285      *   PROCESS 'REF' COMMAND
1286      *
1287      REF
1288      09C0 041E      BLWP *PRNPT   PRINT LINE
1289      REFF
1290      09C2 0420      BLWP @SCAN    GET SYMBOL
           09C4 0C2C'
1291      09C6 1011      JMP  ARFER    NO SYMBOL - ERROR
1292      09C8 0420      BLWP @SRCSYM SEARCH SYMBOL TABLE
           09CA 0D94'
1293      09CC 1003      JMP  ARF2     NOT IN TABLE - GOOD
1294      09CE 0420      BLWP @ER3    ILLEGAL EXTERNAL REFERENCE
           09D0 0B52'
1295      09D2 1008      JMP  ANXTFL   GET NEXT FIELD
1296      ARF2
1297      09D4 C020      MOV  @SYMBOL+NWSCT+NWSCT,NWVAL
           09D6 0D3E'
1298      09D8 0220      AI   NWVAL,8  ADJUST FLAG VALUE

```

```

1299 09DA 0008
1300 09DC C220      MOV  @SYMBOL+SYMPTR+SYMPTR, SPTR
1301 09DE 0D32'
1302 09E0 06C0      SWPB NVAL          PLACE IN SYMBOL TABLE ENTRY
1303 09E2 D600      MOVB NVAL, *SPTR
1304 09E4 06A0      ANXTFL           BL  @NXTFLD        CHECK NEXT FIELD
1305 09E6 0ACC'
1306 09E8 10EC      JMP  REFF        MORE SYMBOLS - CONTINUE
1307 09EA 0420      ARFER           BLWP @ER2
1308 09EC 0B4E'
1309 09EE 10FA      JMP  ANXTFL      TEST NEXT FIELD
1310 *
1311 * PROCESS 'RORG' COMMAND
1312 *
1313 ARORG
1314 09F0 05A0      INC  @OTWP+NCONS+NCONS SET NON-CONSECUTIVE FLAG
1315 09F2 16E6'
1316 09F4 C160      MOV  @DORGFL, ZCK IS CODE PRESENTLY IN DORG?
1317 09F6 11C8'
1318 09F8 1602      JNE  ARORG2      YES
1319 09FA C0C3      MOV  RELABS, RELABS PRESENTLY RELOCATABLE?
1320 09FC 1606      JNE  RELORG      YES, SKIP
1321 09FE 0203      ARORG2 LI RELABS, 1 MARK CODE AS RELOCATABLE
1322 0A00 0001
1323 0A02 C120      MOV  @RELPC, PC  GET OLD RELOC VALUE
1324 0A04 0C18'
1325 0A06 04E0      CLR  @DORGFL    RESET DORG FLAG
1326 0A08 11C8'
1327 REORG
1328 0A0A 0420      BLWP @LKCHAR    LOOK AT NEXT CHARACTER
1329 0A0C 115C'
1330 0A0E 0282      CI  CHAR, >D   IF CARRIAGE RETURN, RETURN
1331 0A10 000D
1332 0A12 130A      JEQ  ROUT
1333 0A14 06A0      BL  @WDE        GET WELL DEFINED EXPRESSION
1334 0A16 0AC0'
1335 0A18 C160      MOV  @EXWP+RELVL+RELVL, ZCK EXPRESSION RELOCATABLE?
1336 0A1A 0F28'
1337 0A1C 1603      JNE  ARORG3      YES
1338 0A1E 0420      BLWP @ER2        NO - SYNTAX ERROR
1339 0A20 0B4E'
1340 0A22 1002      JMP  ROUT
1341 ARORG3
1342 0A24 C120      MOV  @EXWP+RESLT+RESLT, PC REPLACE OLD PC
1343 0A26 0F22'
1344 ROUT
1345 0A28 C160      MOV  @LABSN, ZCK WAS LABEL PRESENT
1346 0A2A 15A6'
1347 0A2C 1309      JEQ  ROUT2       NO - EXIT
1348 0A2E C1E0      MOV  @LABPTR, NWSYM YES - POINT TO IT
1349 0A30 0068'
1350 0A32 55E0      SZCB @ROAB, *NWSYM ADJUST RELOCATABILITY
1351 0A34 006F'

```



```

1336 0A36 C807      MOV  NWSYM, @SYMBOL+SYMPTR+SYMPTR
      0A38 0D32'
1337 0A3A 041F      BLWP *GVLPT          POINT TO SYMBOL VALUE
1338 0A3C C19D      MOV  *SYMNPT, SNAM   ADJUST VALUE
1339 0A3E C584      MOV  PC, *SNAM
1340
1341 0A40 06A0      BL   @PCPRN          PLACE PC IN OUTPUT LINE
      0A42 1490'
1342
1343 0A44 0460      B    @APRLN          PRINT LINE
      0A46 0542'
1344
1345 0A48 0460      B    @SYNTAX
      0A4A 0790'
1346
1347
1348
1349
1350 0A4C 040A      CLR  SFLAGS          CLEAR NEGATIVE FLAG
1351 0A4E 0411      BLWP *GETC           GET CHARACTER
1352 0A50 0282      CI   CHAR, >2D      IS IT A MINUS
      0A52 002D
1353 0A54 1602      JNE  NOM             NO MINUS
1354 0A56 058A      INC  SFLAGS          YES, SET NEGATIVE FLAG
1355 0A58 0411      BLWP *GETC           GET CHARACTER
1356
1357 0A5A 0282      CI   CHAR, >27      IS CHAR A QUOTE
      0A5C 0027
1358 0A5E 16F4      JNE  SYNTAX2        NO, SYNTAX ERROR
1359 0A60 05A0      INC  @SCWP+INCSC+INCSC SET INSIDE STRING FLAG
      0A62 0C1C'
1360
1361 0A64 0420      BLWP @SCAN           GET QUOTED PAIR
      0A66 0C2C'
1362 0A68 1000      NOP
1363 0A6A C020      MOV  @SCWP+RESLT+RESLT, NWVAL
      0A6C 0C12'
1364 0A6E 06C0      SWPB NWVAL
1365 0A70 D000      MOVB NWVAL, NWVAL   IS THERE ONLY ONE CHARACTER
1366 0A72 130C      JEQ  EXT0            YES, LAST CHARACTER
1367 0A74 06A0      BL   @OUTNW2        NO - OUTPUT VALUE
      0A76 0AE8'
1368 0A78 0620      DEC  @SCWP+SCPT+SCPT DECREMENT SCAN POINTER
      0A7A 0C16'
1369 0A7C 0420      BLWP @LKCHAR        WAS SEC CHAR A QUOTE
      0A7E 115C'
1370 0A80 0282      CI   CHAR, >27
      0A82 0027
1371 0A84 16EF      JNE  GT0            NO, CONTINUE
1372 0A86 0620      DEC  @SCWP+SCPT+SCPT YES, DECREMENT SCAN POINTER
      0A88 0C16'
1373 0A8A 10EC      JMP  GT0            GET NEXT QUOTED PAIR
1374
1375 0A8C C28A      MOV  SFLAGS, SFLAGS IS NEGATE SET
1376 0A8E 1301      JEQ  EXT0            NO - EXIT
      EXT0'
    
```

1377	0A90	0500	NEG	NWVAL	YES - NEGATE LAST BYTE
1378			EXTQ2		
1379	0A92	06A0	BL	@OUTNW2	OUTPUT VALUE
	0A94	0AE8			
1380	0A96	0411	BLWP	*GETC	GET CHARACTER
1381	0A98	0411	BLWP	*GETC	GET NEXT CHARACTER
1382	0A9A	0282	CI	CHAR, >20	IS CHARACTER A BLANK
	0A9C	0020			
1383	0A9E	1302	JEQ	TXTEND	YES - END OF TEXT
1384	0AA0	0420	BLWP	@ER2	NO - SYNTAX ERROR
	0AA2	0B4E			
1385			TXTEND		
1386	0AA4	0460	B	@AREAD	READ NEXT RECORD
	0AA6	050A			
1387			*		
1388			*	PROCESS 'UNL' COMMAND	
1389			*		
1390			AUNL		
1391	0AA8	04E0	CLR	@LIST	CLEAR LIST FLAG
	0AAA	1856			
1392	0AAC	10FB	JMP	TXTEND	READ NEXT RECORD
1393			*		
1394			*	EVALUATE EXPRESSION	
1395			*		
1396			EXEVL		
1397	0AAE	C24B	MOV	11, NULL	SAVE RETURN
1398	0AB0	06A0	BL	@WDE	GET WELL DEFINED EXPR
	0AB2	0AC0			
1399	0AB4	C160	MOV	@EXWP+RELVL+RELVL, ZCK	EXPRESSION RELOCATABLE?
	0AB6	0F28			
1400	0AB8	16C7	JNE	SYNTAX2	EXPRESSION IS RELOC ERROR
1401	0ABA	A120	A	@EXWP+RESLT+RESLT, PC	PLACE IN PC
	0ABC	0F22			
1402	0ABE	0459	B	*NULL	RETURN
1403			*		
1404			*	GET WELL-DEFINED EXPRESSION	
1405			*		
1406			WDE		
1407	0AC0	05A0	INC	@NFWDA	MARK NO FWD ALLOWD
	0AC2	11B8			
1408	0AC4	0420	BLWP	@EXPR	GET EXPRESSION
	0AC6	0F42			
1409	0AC8	1000	NOP		FORWARD REF RETURN
1410	0ACA	045B	RT		RETURN
1411			*		
1412			*	CHECK NEXT FIELD	
1413			*		
1414			NXTFLD		
1415	0ACC	0411	BLWP	*GETC	GET NEXT CHARACTER
1416	0ACE	0282	CI	CHAR, >20	IS CHAR A BLANK
	0AD0	0020			
1417	0AD2	13E8	JEQ	TXTEND	YES - READ NEXT RECORD
1418	0AD4	0282	CI	CHAR, >20	IS CHAR A COMMA
	0AD6	0020			
1419	0AD8	1306	JEQ	NXOUT	YES - CONTINUE

```

1420 0ADA 0420      BLWP @ER2      SYNTAX ERROR
      0ADC 0B4E'
1421 0ADE C160      MOV  @FSTPT,ZCK  HAS LINE BEEN PRINTED
      0AE0 1858'
1422 0AE2 13E0      JEQ  TXTEND      YES - READ RECORD
1423 0AE4 10AF      JMP  PRG4        NO - PRINT LINE
1424
1425 0AE6 045B      NXOUT          RT      RETURN
1426
1427          *      OUTPUT TEXT VALUE
1428          *
1429          OUTNW2
1430 0AE8 0420      BLWP @BINHEX    PUT OUT VALUE
      0AEA 0BE0'
1431 0AEC 0428'      DATA NWVL      BYTE VALUE
1432 0AEE 0138'      DATA INSVL     LOCATION IN OUTPUT LINE
1433 0AF0 C820      MOV  @BLANK,@INSVAL  BLANK OUT FIRST BYTE
      0AF2 17DC'
      0AF4 0138'
1434 0AF6 C30B      MOV  11,TYPE    SAVE RETURN
1435 0AF8 06A0      BL   @PCPRN     PLACE PC IN OUTPUT LINE
      0AFA 1490'
1436 0AFC 0584      INC  PC         PUT OUT AND INCR LOC
1437 0AFE C804      MOV  PC,@PPC   ADJUST PRESENT PC
      0B00 11BA'
1438 0B02 0420      BLWP @OUTBYT    OUTPUT BYTE
      0B04 17E8'
1439 0B06 C160      MOV  @FSTPT,ZCK  FIRST TIME FOR PRINT?
      0B08 1858'
1440 0B0A 1301      JEQ  OUTNW      NO, SKIP
1441 0B0C 041E      BLWP *PRNPT     PRINT LINE
1442          OUTNW
1443 0B0E 045C      B    *TYPE      RETURN
1444          *
1445          *      MAKE PROGRAM COUNTER EVEN
1446          *
1447          REVEN
1448 0B10 0584      INC  PC         MAKE PROGRAM COUNTER EVEN
1449 0B12 0244      ANDI PC,>FFFE
      0B14 FFFE
1450 0B16 C160      MOV  @LABSN,ZCK  IS LABEL PRESENT
      0B18 15A6'
1451 0B1A 1306      JEQ  REVOUT     NO - QUIT
1452 0B1C C820      MOV  @LABPTR,@SYMBOL+SYMPTR+SYMPTR  YES:
      0B1E 0068'
      0B20 0D32'
1453 0B22 041F      BLWP *GVLPT     POINT TO SYMBOL VALUE
1454 0B24 C19D      MOV  *SYMNPT,SNAM  ADJUST VALUE IN LABEL
1455 0B26 C584      MOV  PC,*SNAM
1456          REVOUT
1457 0B28 04E0      CLR  @OTBWP+BYTFL+BYTFL  CLEAR BYTE BOUNDARY FLAG
      0B2A 17D8'
1458 0B2C 045B      RT      RETURN
    
```

```
1461 *  
1462 * ERROR PROCESSOR WORKSPACE AREA  
1463 *  
1464 0B2E 0000 LPWP DATA 0 R0 ERRG  
1465 0B30 0020 DATA >20 R1 BLNKK  
1466 0B32 0000 DATA 0 R2 CHAR  
1467 0B34 0008 DATA ERMSG R3 MSGPT  
1468 0B36 001E DATA LERADD R4 LMSGPT  
1469 0B38 0006 ENDST DATA SYMT R5  
1470 0B3A 0C16 DATA SCWP+SCPT+SCPT R6 SCNR  
1471 0B3C 0BF4 DATA BINDEC R7 BINDPT  
1472 0B3E 3200 DATA >3200 R8 EROFST  
1473 0B40 3100 ZZ1 DATA >3100 R9  
1474 0B42 0000 DATA 0 R10 IOPARM  
1475 0B44 0000 NUMERR DATA 0 R11 NUMERG  
1476 0B46 15AE DATA PGHDR R12 PGP  
1477 0B48 0000 DATA 0 R13 OLD WP  
1478 0B4A 0000 DATA 0 R14 OLD PC  
1479 0B4C 0000 DATA 0 R15 OLD SR
```

```
1481 * TITLE: ERROR
1482 * ERROR PROCESSOR
1483 * REVISION: 03/01/74
1484 * ORIGINAL
1485 * 03/15/76
1486 * MODIFIED TO RUN WITH PXRMTN
1487 * COMPUTER: 990, ASSEMBLY
1488 * ABSTRACT:
1489 * SEVEN ENTRY POINTS ARE GIVEN INTO THE ERROR
1490 * PROCESSOR TO CREATE THE APPROPRIATE VALUE:
1491 * ER2 - SYNTAX ERROR
1492 * ER3 - ILLEGAL EXTERNAL REFERENCE
1493 * ER4 - TRUNCATION ERROR
1494 * ER5 - MULTIPLY DEFINED SYMBOL
1495 * ER6 - UNRECOGNIZABLE OPERATOR
1496 * ER7 - ILLEGAL FORWARD REFERENCE
1497 * ER8 - ILLEGAL TERM
1498 * THE PROCESSOR WILL SCAN THE INPUT LINE TO THE
1499 * NEXT COMMA, BLANK, OR EOL TO ATTEMPT TO
1500 * CONTINUE SCAN OF LINE.
1501 * CALLING SEQUENCE:
1502 * BLWP @ERX X = 2, 3, 4, 5, 6, 7, 8
1503 * ADDITIONAL NOTES:
1504 * WORKSPACE = LPWP(UNSHARED)
1505 * ROUTINES CALLED: BINDEC, PGHDR, PRINT
1506 ER2
1507 0B4E 0B2E' DATA LPWP
1508 0B50 0B76' DATA ERROR2 SYNTAX ERROR
1509 ER3
1510 0B52 0B2E' DATA LPWP
1511 0B54 0B74' DATA ERROR3 ILLEGAL EXTERNAL REFERENCE
1512 ER4
1513 0B56 0B2E' DATA LPWP
1514 0B58 0B72' DATA ERROR4 TRUNCATION ERROR
1515 ER5
1516 0B5A 0B2E' DATA LPWP
1517 0B5C 0B70' DATA ERROR5 MULTIPLY DEFINED SYMBOL ERROR
1518 ER6
1519 0B5E 0B2E' DATA LPWP
1520 0B60 0B6E' DATA ERROR6 UNRECOGNIZABLE OPERATOR ERROR
1521 ER7
1522 0B62 0B2E' DATA LPWP
1523 0B64 0B6C' DATA ERROR7 ILLEGAL FORWARD REFERENCE
1524 ER8
1525 0B66 0B2E' DATA LPWP
1526 0B68 0B6A' DATA ERROR8 ILLEGAL TERM
1527 *
1528 * ENTRY POINTS
1529 *
1530 ERROR8
1531 0B6A 0580 INC ERRG
1532 ERROR7
1533 0B6C 0580 INC ERRG
1534 ERROR6
```

1535	0B6E	0580	INC	ERRG	
1536			ERROR5		
1537	0B70	0580	INC	ERRG	
1538			ERROR4		
1539	0B72	0580	INC	ERRG	
1540			ERROR3		
1541	0B74	0580	INC	ERRG	
1542			ERROR2		
1543			*		
1544			*	CREATE ERROR MESSAGE	
1545			*		
1546	0B76	0A00	SLA	ERRG, 8	SHIRT ERROR NUMBER LEFT
1547	0B78	A008	A	EROFST, ERRG	CONVERT FOR ASCII
1548	0B7A	D800	MOVB	ERRG, @ERNUM	PLACE ERROR NUMBER IN OUTPUT
	0B7C	0011			
1549	0B7E	C820	MOV	@RCDCNT, @BINVL	CONVERT STATEMENT NUMBER
	0B80	1862			
	0B82	0BC0			
1550	0B84	0417	BLWP	*BINDPT	TO DECIMAL AND PLACE
1551	0B86	001A	DATA	ERLOC	IN ERROR MESSAGE
1552	0B88	0620	DEC	@LNCNT	DECR LINE COUNT OF PRINT
	0B8A	1590			
1553	0B8C	1501	JGT	MSGPRT	IF ROOM, PRINT MESSAGE
1554	0B8E	041C	BLWP	*PGP	NO ROOM, PRINT HEADER
1555			MSGPRT		
1556	0B90	C283	MOV	MSGPT, IOPARM	POINT TO ERROR MESSAGE
1557	0B92	0420	BLWP	@PRINT	PRINT MESSAGE
	0B94	19CE			
1558	0B96	C520	MOV	@BLANK, *LMSGPT	EXTEND TO INCL LAST ERR MSG
	0B98	17DC			
1559	0B9A	C820	MOV	@RCDCNT, @BINVL	CONVERT CURRENT RECORD COUNT
	0B9C	1862			
	0B9E	0BC0			
1560	0BA0	0417	BLWP	*BINDPT	TO DECIMAL AND STORE
1561	0BA2	0030	DATA	LERLOC	IN LAST ERROR MESSAGE
1562	0BA4	058B	INC	NUMERG	INCR NUMBER OF ERRORS
1563			LPBG		
1564	0BA6	0420	BLWP	@GETCHR	GET CHARACTER
	0BA8	116E			
1565	0BAA	8042	C	CHAR, BLNKK	IS IT A BLANK
1566	0BAC	1306	JEQ	EXLP	YES, EXIT
1567	0BAE	0282	CI	CHAR, >2C	A COMMA?
	0BB0	002C			
1568	0BB2	1303	JEQ	EXLP	YES, EXIT
1569	0BB4	0282	CI	CHAR, EOL	END OF LINE?
	0BB6	000D			
1570	0BB8	16F6	JNE	LPBG	NO, CONTINUE SCANNING
1571			EXLP		
1572	0BBA	0616	DEC	*SCNR	POINT TO LAST CHARACTER READ
1573	0BBC	04C0	CLR	ERRG	CLEAR ERROR TYPE
1574	0BBE	0380	RTWP		RETURN

```

1577 *
1578 * CONVERSION ROUTINE WORKSPACE AREA
1579 *
1580 CONVWS
1581 0BC0 0000 BINVL DATA 0 R0 BINVAL BINARY VALUE
1582 0BC2 0000 DATA 0 R1 VALSV TEMP VALUE SAVE
1583 0BC4 0000 DATA 0 R2 MADD LOCN ASCII VALUE
1584 0BC6 0000 DATA 0 R3 SCCODE SUPV CALL CODE
1585 0BC8 0000 ASCVL1 DATA 0 R4 ASCII VALUE
1586 0BCA 0000 ASCVL2 DATA 0 R5
1587 0BCC 0000 DATA 0 R6
1588 0BCE 0C00 DATA >C00 R7 BHCODE BINARY/HEX SCC
1589 0BD0 0A00 DATA >A00 R8 BDCODE BINARY/DECIMAL SCC
1590 0BD2 FFFC DATA -4 R9 HCNT
1591 0BD4 0000 DATA 0 R10 PCNT CHAR MOVE COUNT
1592 0BD6 0085' OBJSTP DATA OBJST R11
1593 0BD8 0078' OBJBUF DATA OBJRCD R12
1594 0BDA 0000 DATA 0 R13 OLD WP
1595 0BDC 0000 DATA 0 R14 CHOPC OLD PC
1596 0BDE 0000 DATA 0 R15 OLD ST
    
```

```

1598 * TITLE: CONVRT (BINHEX/BINDEC)
1599 *
1600 * REVISION: 03/01/74
1601 *
1602 * ORIGINAL
1603 *
1604 * 03/15/76
1605 * MODIFIED TO RUN WITH PXRMTN
1606 * COMPUTER: 990, ASSEMBLY
1607 * ABSTRACT: SETS UP SUPERVISOR CALL BLOCK FOR CALL
1608 * AND STORES VALUES AFTER RETURN IN USER
1609 * DEFINED LOCATIONS.
1610 * CALLING SEQUENCE:
1611 * BINARY TO HEX ASCII
1612 * BLWP @BINHEX
1613 * DATA ADDRESS-OF-VALUE
1614 * DATA ADDRESS-FOR-PLACEMENT
1615 * BINARY TO DECIMAL ASCII
1616 * MOV VALUE,@BINVL
1617 * BLWP @BINDEC
1618 * DATA ADDRESS-FOR-PLACEMENT
1619 BINHEX
1620 0BE0 0BC0' DATA CONVMS
1621 0BE2 0BE4' DATA BIHEX
1622 BIHEX
1623 0BE4 C03E MOV *CHOPC+,BINVAL POINT TO ADDRESS OF VALUE
1624 0BE6 C010 MOV *BINVAL,BINVAL GET VALUE IN R0 FOR CALL
1625 0BE8 C0C7 MOV BHCODE,SCCODE SET UP SUPV CALL CODE
1626 0BEA 2FC3 SVC SCCODE BINARY TO HEX SUPV CALL
1627 0BEC C0BE MOV *CHOPC+,MADD GET PLACEMENT LOC & NEW RTN
1628 0BEE 0201 LI VALSV,ASCVL1 SET UP ASCII VALUE POINTER
1629 0BF0 0BC8'
1630 0BF2 1007 JMP NOVAL
1631 *
1632 BINDEC
1633 0BF4 0BC0' DATA CONVMS
1634 0BF6 0BF8' DATA BINDC
1635 *
1636 BINDC
1637 0BF8 C0C8 MOV BDCODE,SCCODE SET UP SUPERVISOR CALL CODE
1638 0BFA 2FC3 SVC SCCODE BINARY TO DECIMAL SUPV CALL
1639 0BFC C0BE MOV *CHOPC+,MADD GET PLACEMENT LOC & NEW RTN
1640 0BFE 0201 LI VALSV,ASCVL2 SET UP ASCII VALUE POINTER
1641 0C00 0BCA'
1642 MOVAL
1643 0C02 C289 MOV HCNT,PCNT INIT COUNT TO 4
1644 BIN
1645 0C04 DCB1 MOV *VALSV+,*MADD+ MOVE CONVERTED DECIMAL VALUE
1646 0C06 058A INC PCNT TO USER SPECIFIED LOCATION
1647 0C08 11FD JLT BIN
1648 0C0A 0380 RTWP RETURN
  
```



```
1647 *
1648 * SCAN WORKSPACE AREA
1649 *
1650 SCWP
1651 0C0C 0008 DATA 8 WR0 QTCLS
1652 0C0E 0000 DATA 0 WR1 CLSTYP
1653 0C10 0000 DATA 0 WR2 CHAR
1654 0C12 0000 DATA 0 WR3 RESLT
1655 0C14 0B56 DATA ER4 WR4 ERR4
1656 0C16 0000 DATA 0 WR5 SCPT
1657 0C18 0000 RELPC DATA 0
1658 0C1A 0CE4 DATA NXCHAR WR7 NXCPT
1659 0C1C 0000 DATA 0 WR8 INCSC
1660 0C1E 0000 DATA 0 WR9 SCNT
1661 0C20 0D18 DATA TQUOT WR10 TQC
1662 0C22 0000 DATA 0 WR11 RTREG
1663 0C24 0000 DATA 0 WR12 SAVE
1664 0C26 0000 DATA 0 WR13 OLD WP
1665 0C28 0000 DATA 0 WR14 OLD PC
1666 0C2A 0000 DATA 0 WR15 OLD SR
```

```

1668      * TITLE:      SCAN
1669      *              SCAN PROCESSING
1670      * REVISION:   03/01/74
1671      *              ORIGINAL
1672      * COMPUTER:   990, ASSEMBLY
1673      * ABSTRACT:
1674      *              THE FIRST CHARACTER IS USED TO DETERMINE IF THE
1675      *              SCAN VALUE IS A SYMBOL OR A NUMERIC CONSTANT.
1676      *              THE SCAN OF A SYMBOL CONTINUES UNTIL A NON-
1677      *              ALPHANUMERIC CHARACTER IS SEEN. BUT ONLY THE
1678      *              FIRST SIX CHARACTERS ARE SAVED (ZERO-FILL). A
1679      *              COUNT OF THE NUMBER OF CHARACTERS IS KEPT.
1680      *              THE SCAN OF A NUMERIC VALUE CONTINUES UNTIL A
1681      *              INAPPROPRIATE CHARACTER IS REACHED. THE
1682      *              NUMERIC VALUE IS CONVERTED INTERNALLY.
1683      * CALLING SEQUENCE:
1684      *              BLWP @SCAN
1685      *              JMP NUMERIC LOCATION JUMP TO NUMERIC PROCESSOR
1686      * ADDITIONAL NOTES:
1687      *              WORKSPACE = SCWP (UNSHARED)
1688      *              TWO POSSIBLE RETURNS MAY BE TAKEN BY SCAN. IF
1689      *              A NUMERIC VALUE IS FOUND, THE CONVERTED NUMERIC
1690      *              VALUE PLACED IN RESULT AND PROCESSING CONTINUES
1691      *              AT THE MEMORY WORD FOLLOWING THE CALL. IF A
1692      *              SYMBOL IS FOUND, RESULT WILL CONTAIN A POINTER
1693      *              TO THE SIX CHARACTER NAME (THIS POINTER WILL
1694      *              ALSO BE GIVEN TO MAINW+NWSYM), THE
1695      *              CHARACTER COUNT IS PLACED IN SYMBOL+NWSCT+NWSCT
1696      *              AND PROCESSING CONTINUES AT THE SECOND WORD
1697      *              AFTER THE CALL SCAN UTILIZES A CLASS TABLE IN
1698      *              ANALYZING CHARACTERS.
1699      *              ROUTINES CALLED: ER4, ER2, NXCHAR, TQUOT
1700      * SCAN
1701      0C2C  0C0C'  DATA SCWP      WORKSPACE
1702      0C2E  0C30'  DATA SCBG      START
1703      *
1704      * CLASSIFY FIRST CHARACTER AND BRANCH
1705      * TO APPROPRIATE PROCESSING CODE
1706      *
1707      * SCBG
1708      0C30  C208    MOV  INCSC, INCSC    INSIDE OF STRING?
1709      0C32  1666    JNE  QUOTE          YES
1710      0C34  0203    LI   RESULT, SYMBL  INITIALIZE RESULT POINTER
1711      0C36  114A'
1712      0C38  0697    BL   *NXCPT         GET NEXT CHARACTER CLASS
1713      0C3A  0221    AI   CLSTYP, -5    ADJUST CLASS TYPE
1714      0C3C  FFFB
1715      0C3E  114D    JLT  ERRORS        TOO SMALL - ERROR
1716      0C40  D821    MOVB @JMPTB1(CLSTYP), @JMP1+1
1717      0C42  1A2E'
1718      0C44  0C47'
1719      0C46  10FF    JMP1  JMP  #        BRANCH ON CHARACTER CLASS
1720      *
1721      * SYMBOL PROCESSOR
  
```

```

1718
1719      *
1720      ALPHA
1720 0048 04F3 CLR *RESLT+ CLEAR RESULT
1721 004A 04F3 CLR *RESLT+
1722 004C 04D3 CLR *RESLT
1723 004E 0223 AI RESLT, -4 RESET POINTER
      0050 FFFC
1724 0052 0209 LI SCNT, 5 SET COUNT
      0054 0005
1725      MOVCHR
1726 0056 0A82 SLA CHAR, 8 POSITION CHARACTER
1727 0058 DCC2 MOVB CHAR, *RESLT+ MOVE INTO SYMBOL NAME
1728      CONGET
1729 005A 0697 BL *NXCPT GET CHARACTER CLASS
1730 005C 0281 CI CLSTYP, 5 IS CHARACTER ALPHANUMERIC
      005E 0005
1731 0060 1106 JLT NOADD NO - OUT OF RANGE
1732 0062 0281 CI CLSTYP, 7 NO - OUT OF RANGE
      0064 0007
1733 0066 1503 JGT NOADD NO - OUT OF RANGE
1734 0068 0609 DEC SCNT YES, TEST FOR ADDING
1735 006A 11F7 JLT CONGET DON'T ADD
1736 006C 10F4 JMP MOVCHR ADD
1737      NOADD
1738 006E 0605 DEC SCPT DECREMENT SCAN POINTER
1739 0070 0249 MOV SCNT, SCNT IS COUNT > 6
1740 0072 1501 JGT MVCNT NO
1741 0074 04C9 CLR SCNT CLEAR COUNT
1742      MVCNT
1743 0076 0509 NEG SCNT SET PROPER CHARACTER COUNT
1744 0078 0229 AI SCNT, 6
      007A 0006
1745 007C 0809 MOV SCNT, @SYMBOL+NWSCT+NWSCT PASS TO SYMBOL
      007E 0D3E
1746 0080 0203 LI RESLT, SYMBL POINT TO SYMBOL NAME
      0082 114A
1747 0084 0803 MOV RESLT, @MAINW+NWSYM+NWSYM PASS TO MAINW
      0086 0436
1748 0088 05CE INCT 14 TAKE ALTERNATE RETURN
1749 008A 0380 RTWP RETURN
1750      *
1751      * DECIMAL INTEGER PROCESSOR
1752      *
1753      NUM
1754 008C 00C2 MOV CHAR, RESLT GET FIRST DIGIT
1755 008E 0223 AI RESLT, ->30 ADJUST FOR DECIMAL VALUE
      0090 FFD0
1756      NXGET
1757 0092 0697 BL *NXCPT GET NEXT CHARACTER CLASS
1758 0094 0281 CI CLSTYP, 6 IS CHARACTER NUMERIC
      0096 0006
1759 0098 1302 JEQ NUMAD YES, CONTINUE
1760      NUMD
1761 009A 0605 DEC SCPT NO, DECREMENT SCAN POINTER
1762 009C 0380 RTWP RETURN
  
```

```

1763
1764 0C9E 0222 NUMAD AI CHAR, ->30 YES, ADJUST FOR VALUE
      0CA0 FFD0
1765 0CA2 0A13 SLA RESULT, 1 ADD TO RESULT
1766 0CA4 A083 A RESULT, CHAR
1767 0CA6 0A23 SLA RESULT, 2
1768 0CA8 A0C2 A CHAR, RESULT
1769 0CAA 17F3 JNC NXGET IF NO OVERFLOW, CONTINUE
1770 0CAC 0414 BLWP *ERR4 TRUNCATION ERROR
1771 0CAE 10F1 JMP NXGET GET NEXT DIGIT
1772 *
1773 * HEXADECIMAL INTEGER PROCESSOR
1774 *
1775 HEX
1776 0CB0 04C3 CLR RESULT CLEAR RESULT
1777 HEXX
1778 0CB2 0697 BL *NXOPT GET NEXT CHARACTER CLASS
1779 0CB4 0221 AI CLSTYP, -5 ALPHANUMERIC?
      0CB6 FFFB
1780 0CB8 11F0 JLT NUM0 NOT ALPHANUMERIC
1781 0CBA D821 MOVB @JMPTB2(CLSTYP), @JMP2+1
      0CBC 1A33
      0CBE 0CC1
1782 0CC0 10FF JMP2 JMP $ USE JUMP TABLE
1783 AL
1784 0CC2 0282 CI CHAR, >46 OUT OF RANGE?
      0CC4 0046
1785 0CC6 15E9 JGT NUM0 YES - EXIT
1786 0CC8 0222 AI CHAR, -7 ADJUST FOR ALPHABETIC
      0CCA FFF9
1787 NU
1788 0CCC 0222 AI CHAR, ->30 ADJUST FOR ASCII
      0CCE FFD0
1789 0CD0 0A43 SLA RESULT, 4 SHIFT CHARACTER VALUE
1790 0CD2 1701 JNC ACHAR IF NO OVERFLOW, CONTINUE
1791 0CD4 0414 BLWP *ERR4 TRUNCATION ERROR
1792 ACHAR
1793 0CD6 A0C2 A CHAR, RESULT ADD CHARACTER VALUE IN
1794 0CD8 10EC JMP HEXX CONTINUE
1795 ERRORS
1796 0CDA 0605 DEC SCPT DECREMENT SCAN POINTER
1797 0CDC 0420 BLWP @ER2 SYNTAX ERROR
      0CDE 0B4E
1798 0CE0 04C3 CLR RESULT CLEAR RESULT
1799 0CE2 0380 RTWP RETURN
1800 *
1801 * CLASSIFY CHARACTER
1802 *
1803 NXCHAR
1804 0CE4 D0B5 MOVB *SCPT+, CHAR GET CHARACTER
1805 0CE6 0982 SRL CHAR, 8 RIGHT JUSTIFY
1806 0CE8 0282 CI CHAR, >09 TAB?
      0CEA 0009
1807 0CEC 1602 JNE NXCH2 NO
1808 0CEE 0202 LI CHAR, >20 YES - SET TO BLANK
  
```

```

1809      0CF0  0020
1810      0CF2  0282      NXCH2      CI  CHAR, >20      IS CHARACTER LESS THAN >20
          0CF4  0020
1811      0CF6  11F1      JLT  ERRORS      YES
1812      0CF8  D062      MOVB @CLASS->20(CHAR), CLSTYP  GET CLASS
          0CFA  019A
1813      0CFC  0981      SRL  CLSTYP, 8      RIGHT JUSTIFY
1814      0CFE  045B      RT      RETURN
1815
1816      *
1817      * QUOTED STRING PROCESSOR
1818      *
1819      * QUOTE
1819      0D00  0697      BL  *NXCPT      GET NEXT CHARACTER CLASS
1820      0D02  8001      C  CLSTYP, QTCLS  IS THIS A QUOTE
1821      0D04  1601      JNE  NQ      NO
1822      0D06  069A      BL  *TQC      YES - ADDITIONAL QUOTE?
1823
1824      0D08  C0C2      NO
1824      MOV  CHAR, RESULT  NO, MOVE CHARACTER
1825      0D0A  0697      BL  *NXCPT      GET NEXT CHARACTER CLASS
1826      0D0C  8001      C  CLSTYP, QTCLS  TEST FOR QUOTE
1827      0D0E  1601      JNE  NQ2      NO
1828      0D10  069A      BL  *TQC      YES - ADDITIONAL QUOTE?
1829
1830      0D12  0A83      NQ2
1830      SLA  RESULT, 8      LEFT JUSTIFY
1831      0D14  E0C2      SOC  CHAR, RESULT  PLACE IN RESULT
1832      0D16  0380      RTWP      RETURN
1833
1834      0D18  C30B      TQUOT
1834      MOV  RTN, SAVE
1835      0D1A  0697      BL  *NXCPT      GET NEXT CHARACTER CLASS
1836      0D1C  8001      C  CLSTYP, QTCLS  IS IT A QUOTE
1837      0D1E  1302      JEQ  00      YES - EXIT
1838      0D20  0645      DECT SCPT      NO, END OF STRING
1839      0D22  04C2      CLR  CHAR      CLEAR CHARACTER
1840
1841      0D24  045C      00
1841      B  *SAVE      YES, RETURN
1842
1843      *
1844      * DOLLAR SIGN PROCESSOR
1845      *
1846      DOLLAR
1846      0D26  C0E0      MOV  @MAINW+PC+PC, RESULT  GET PC VALUE AND RELOC
          0D28  0430
1847      0D2A  C820      MOV  @MAINW+RELABS+RELABS, @EXWP+RLVAL+RLVAL
          0D2C  042E
          0D2E  0F2A
1848      0D30  0380      RTWP      RETURN

```

1851			*			
1852			* SYMBOL TABLE ENTRY REGION			
1853			*			
1854			SYMBOL			
1855	0D32	0000	DATA 0	R0	SYMPTR	
1856	0D34	0000	DATA 0	R1	SYMNO, COUNT	
1857	0D36	0000	DATA 0	R2	FLAGS	
1858	0D38	0000	DATA 0	R3	SYMNM	
1859	0D3A	0000	DATA 0	R4	NSYM	
1860	0D3C	0000	VL DATA 0	R5	VALUE	
1861	0D3E	0000	DATA 0	R6	NWSCT	
1862	0D40	0000	DATA 0	R7	SCNT2	
1863	0D42	0000	DATA 0	R8	NSBIT	
1864	0D44	0000	DATA 0	R9	GSHOLD	
1865	0D46	0000	DATA 0	R10	GSHLD2, IOPARM	
1866	0D48	0000	DATA 0	R11	RETURN	
1867	0D4A	0000	DATA 0	R12	OUTS	
1868	0D4C	0000	DATA 0	R13	OLD WF	
1869	0D4E	0000	DATA 0	R14	OLD PC	
1870	0D50	0000	DATA 0	R15	OLD SR	

```

1872 * TITLE: SENTRY
1873 * SYMBOL TABLE ENTRY VALUE
1874 * REVISION: 03/01/74
1875 * ORIGINAL
1876 * COMPUTER: 990, ASSEMBLY
1877 * ABSTRACT:
1878 *
1879 * FOUR ROUTINES ARE USED TO PROVIDE:
1880 * 1. CHARACTER COUNT FOR SYMBOL NAME GTSCT
1881 * 2. SYMBOL FLAG VALUE GTSFL
1882 * 3. POINTER TO SYMBOL NAME GTSNM
1883 * 4. POINTER TO VALUE AND VALUE GTSVL
1884 * TWO DUMMY ENTRY POINTS ARE CREATED
1885 * FOR ENTRY FROM OUTSIDE THE WORKSPACE.
1886 * CALLING SEQUENCE:
1887 * INSIDE WORKSPACE:
1888 * BL @GTSCT
1889 * BL @GTSFL
1890 * BL @GTSNM
1891 * BL @GTSVL
1892 * OUTSIDE WORKSPACE:
1893 * BLWP @GTSCT
1894 * BLWP @GTSVL
1895 * ADDITIONAL NOTES:
1896 * WORKSPACE = SYMBOL (SHARED WITH SRCSYM,
1897 * SYMCLR, AND REFDEF)
1898 * GTSNM CALLS GTSCT, GTSVL CALLS GTSNM
1899 * A SYMBOL TABLE NODE APPEARS AS FOLLOWS:
1900 * VALUE VALUE VALUE
1901 * S Y OR S Y OR S Y ETC.
1902 * M B M B M B
1903 * O L O FLAGS FLAGS
1904 * FLAGS
1905 * WHERE FLAGS ARE BROKEN DOWN TO A FLAG VALUE IN
1906 * BITS 8-12 AND A SYMBOL COUNT IN BITS 13-15. THE
1907 * COUNT WILL CAUSE A MODIFICATION OF THE NUMBER
1908 * OF BYTES NEEDED TO STORE THE SYMBOL TABLE ENTRY
1909 * THE SYMBOL NAME AND, THE SYMBOL VALUE WILL
1910 * ALWAYS START ON A WORD BOUNDARY.
1911 * THE FLAG VALUE IS A CODE AS BELOW:
1912 * 0 DEF, REL, EXTDEF, RECDEF
1913 * 1 DEF, REL, RECDEF
1914 * 2 DEF, ABS, EXTDEF, RECDEF
1915 * 3 DEF, ABS, RECDEF
1916 * 4 DEF, REL, EXTDEF
1917 * 5 DEF, REL
1918 * 6 DEF, ABS, EXTDEF
1919 * 7 DEF, ABS
1920 * 8 DXOP
1921 * 9 UNREF, EXTDEF
1922 * 10 REF, EXTDEF
1923 * 11 REF, UNDEF
1924 * 12 EXTREF, REFABS
1925 * 13 EXTREF, REFREL
1926 * WHERE:
    
```

1926			*	DEF	=	DEFINED
1927			*	REL	=	RELOCATABLE VALUE
1928			*	EXTDEF	=	EXTERNALLY DEFINED (FORWARD REFS
1929			*			SATISFIED BUT NOT RESOLVED)
1930			*	ABS	=	ABSOLUTE CODE
1931			*	DXOP	=	DEFINED THROUGH DXOP DIRECTIVE
1932			*	UNREF	=	UNREFERENCE
1933			*	UNDEF	=	UNDEFINED
1934			*	REF	=	REFERENCED
1935			*	EXTREF	=	DECLARED AS EXTERNAL REFERENCE
1936			*	REFABS	=	LAST REFERENCED IN ABSOLUTE CODE
1937			*	REFREL	=	LAST REFERENCED IN RELOCATABLE COD
1938			*			
1939			*	GET SYMBOL FLAGS		
1940			*			
1941				GTSFL		
1942	0D52	0090		MOV *SYMPTR, FLAGS		GET FLAGS WORD
1943	0D54	0832		SRA FLAGS, 3		SHIFT VALUE
1944	0D56	0242		ANDI FLAGS, >1F		SEPARATE OUT FLAGS
	0D58	001F				
1945	0D5A	045B		RT		RETURN
1946			*			
1947			*	GET SYMBOL CHARACTER COUNT		
1948			*			
1949				GETSCT		
1950	0D5C	0D32		DATA SYMBOL		WORKSPACE POINTER
1951	0D5E	0D60		DATA GTSCT2		START
1952				GTSCT2		
1953	0D60	06A0		BL @GTSCT		DUMMY CALL
	0D62	0D66				
1954	0D64	0380		RTWP		RETURN
1955				GTSCT		
1956	0D66	0050		MOV *SYMPTR, COUNT		GET COUNT WORD
1957	0D68	0241		ANDI COUNT, 7		SEPARATE OUT COUNT
	0D6A	0007				
1958	0D6C	045B		RT		RETURN
1959			*			
1960			*	GET SYMBOL VALUE		
1961			*			
1962				GETSVL		
1963	0D6E	0D32		DATA SYMBOL		WORKSPACE
1964	0D70	0D72		DATA GTSVL2		START
1965				GTSVL2		
1966	0D72	06A0		BL @GTSVL		DUMMY CALL
	0D74	0D78				
1967	0D76	0380		RTWP		RETURN
1968				GTSVL		
1969	0D78	024B		MOV 11, GSHOLD		SAVE RETURN
1970	0D7A	06A0		BL @GTSNM		GET SYMBOL NAME
	0D7C	0D84				
1971	0D7E	0643		DECT SYMNM		POINT TO SYMBOL VALUE
1972	0D80	0153		MOV *SYMNM, VALUE		GET SYMBOL VALUE
1973	0D82	0459		B *GSHOLD		RETURN
1974			*			
1975			*	GET SYMBOL NAME		

1976			*		
1977			GTSNM		
1978	0D84	C28B	MOV	11, GSHLD2	SAVE RETURN
1979	0D86	06A0	BL	@GTSCT	GET SYMBOL COUNT
	0D88	0D66			
1980	0D8A	C0C0	MOV	SYMPTR, SYMNM	POINT TO SYMBOL NAME
1981	0D8C	60C1	S	COUNT, SYMNM	POINT TO BYTE OFFSET
1982	0D8E	0243	ANDI	SYMNM, >FFFE	SET AT WORD BOUNDARY
	0D90	FFFE			
1983	0D92	045A	B	*GSHLD2	RETURN

```

1986 * TITLE: SRCSYM
1987 * SEARCH SYMBOL TABLE
1988 * REVISION: 03/01/74
1989 * ORIGINAL
1990 * COMPUTER: 990, ASSEMBLY
1991 * ABSTRACT:
1992 * THE SYMBOL TABLE IS SEARCHED FOR A GIVEN
1993 * SYMBOL NAME. IF THE NAME IS NOT FOUND, IT IS
1994 * ADDED TO THE SYMBOL TABLE WITH THE CODE
1995 * REFERENCED BUT UNDEFINED FOR ITS FLAG VALUE.
1996 * IF THE SYMBOL IS FOUND, AN ALTERNATE EXIT IS
1997 * TAKEN. DUE TO THE VARIABLE LENGTH ENTRIES OF
1998 * THE SYMBOL TABLE, IT IS SEARCHED IN A LINEAR
1999 * FASHION, ENDING WHEN THE COUNT REACHES ZERO.
2000 * CALLING SEQUENCE:
2001 * BLWP @SRCSYM
2002 * SYMBOL-NOT-FOUND INSTRUCTION
2003 * SYMBOL-FOUND INSTRUCTION
2004 * ADDITIONAL NOTES:
2005 * WORKSPACE = SYMBOL (SHARED WITH SENTRY,
2006 * SYMCLR, REFDEF)
2007 * WORKSPACE REGISTER NSBIT IS USED IN FORWARD REF
2008 * TABLE ENTRIES. WORKSPACE REGISTER NWSCT HAS
2009 * BEEN PREVIOUSLY SET BY SCAN.
2010 SRCSYM
2011 0D94 0D32' DATA SYMBOL WORKSPACE
2012 0D96 0D98' DATA SRC START
2013 SRC
2014 0D98 0208 LJ NSBIT,1 INITIALIZE SYMBOL NUMBER
2015 0D9A 0001
2015 *
2016 * POINT TO BEGINNING OF SYMBOL TABLE
2017 *
2018 0D9C C020 MOV @SYMTAB, SYMPTR POINT TO SYMBOL TABLE
2019 0D9E 1154'
2019 *
2020 *
2021 * COMPARE TO ENTRY
2022 *
2023 SYMCHK
2024 0DA0 C120 MOV @NSMBL, NSYM POINT TO START OF NEW SYMBOL
2025 0DA2 0436'
2025 0DA4 8800 C SYMPTR, @ENDST IS TABLE ENDED
2026 0DA6 0B38'
2026 0DA8 1212 JLE EXITNO YES - NOT FOUND
2027 0DAA 06A0 BL @GTSNM NO - GET ENTRY
2028 0DAC 0D84'
2028 0DAE 8181 C SYMNO, NWSCT COMPARE COUNTS
2029 0DB0 1608 JNE NOTSAM NOT THE SAME
2030 COMPAR
2031 0DB2 9D33 CB *SYMNM+, *NSYM+ ARE CHARACTERS THE SAME
2032 0DB4 1606 JNE NOTSAM NO
2033 0DB6 0601 DEC SYMNO ARE CHARACTERS FINISHED
2034 0DB8 16FC JNE COMPAR NO, KEEP COMPARING
  
```

2035	0DBA	06A0	BL	@GTSFL	FOUND - GET FLAGS
	0DBC	0D52			
2036	0DBE	05CE	INCT	14	TAKE ALTERNATE RETURN
2037	0DC0	0380	RTWP		RETURN
2038			*		
2039			*		
2040			*	LOOK AT NEXT SYMBOL	
2041			*		
2042				NOTSAM	
2043	0DC2	06A0	BL	@GTSVL	POINT TO LAST WORD OF ENTRY
	0DC4	0D78			
2044	0DC6	0603	DEC	SYMMN	POINT TO FIRST BYTE OF NEXT
2045	0DC8	C003	MOV	SYMMN, SYMPTR	SET AS POINTER
2046	0DCA	0588	INC	NSBIT	INCREMENT SYMBOL NUMBER
2047	0DCC	10E9	JMP	SYMCHK	GET NEXT SYMBOL
2048			*		
2049			*	CREATE NEW ENTRY	
2050			*		
2051				EXITNO	
2052	0DCE	0226	RI	NWSCT, >58	SET FLAG VALUE
	0DD0	0058			
2053	0DD2	0202	LI	FLAGS, 11	
	0DD4	000B			
2054	0DD6	C406	MOV	NWSCT, *SYMPTR	MOVE INTO NEW ENTRY
2055	0DD8	06A0	BL	@GTSNM	POINT TO START OF NAME
	0DDA	0D84			
2056	0DDC	C120	MOV	@NSMBL, NSYM	POINT TO NAME
	0DDE	0436			
2057				END2	
2058	0DE0	DCF4	MOV	*NSYM+, *SYMMN+	MOVE IN NAME
2059	0DE2	0601	DEC	SYMNO	
2060	0DE4	15FD	JGT	END2	
2061	0DE6	06A0	BL	@GTSVL	POINT TO LAST WORD IN ENTRY
	0DE8	0D78			
2062	0DEA	0603	DEC	SYMMN	POINT TO NEXT ENTRY
2063	0DEC	C003	MOV	SYMMN, @ENDST	SET AS END OF TABLE
	0DEE	0B38			
2064	0DF0	06A0	BL	@LIMCHK	CHECK OVERFLOW LIMITS
	0DF2	1192			
2065	0DF4	0380	RTWP		RETURN

```

2068 * TITLE: SYMCLR
2069 * CLEAR FORWARD REFERENCED SYMBOLS
2070 * REVISION: 03/01/74
2071 * ORIGINAL
2072 * COMPUTER: 990, ASSEMBLY
2073 * ABSTRACT: THE SYMBOL TABLE IS SEARCHED FOR RECENTLY
2074 * DEFINED SYMBOLS, I. E., SYMBOLS WHICH HAD
2075 * BEEN FORWARD REFERENCED BUT WHOSE REFERENCES
2076 * HAVE NOT, AS YET, BEEN RESOLVED. FOR EACH
2077 * SUCH SYMBOL, IT CALLS FIXFRF TO RESOLVE ITS
2078 * REFERENCES AND THEN MODIFIES ITS SYMBOL TABLE
2079 * FLAG CODE SO THAT IT IS NO LONGER CONSIDERED
2080 * AS RECENTLY DEFINED.
2081 * CALLING SEQUENCE:
2082 * BLWP @SYMCLR
2083 * ADDITIONAL NOTES:
2084 * WORKSPACE = SYMBOL (SHARED WITH SENTRY, SRCSYM,
2085 * AND REFDEF)
2086 * THIS ROUTINE IS ONLY CALLED WHEN THE FLAG
2087 * NEWFRF IS SET SIGNIFYING THE EXISTENCE OF
2088 * A RECENTLY DEFINED SYMBOL IN THE SYMBOL
2089 * TABLE AND WHEN THE FLAG REGISTER BYTFL OF
2090 * WORKSPACE OTBWP IS CLEAR, SIGNIFYING THAT
2091 * THE OBJECT CODE LAST PUT OUT COMPLETED A
2092 * WORD OF CODE. THESE TWO FLAGS ARE CHECKED
2093 * EACH TIME BEFORE A RECORD IS READ. THE
2094 * COUNT IN NSBIT IS USED AS THE SYMBOL NUMBER
2095 * FOR COMPARISON WITH ENTRIES IN THE FORWARD
2096 * REFERENCE TABLE.
2097 * ROUTINES CALLED: GTSVL, GTSFL, FIXFRF
2098 SYMCLR
2099 0DF6 0D32' DATA SYMBOL WORKSPACE
2100 0DF8 0DFA' DATA SYMCBG START
2101 SYMCBG
2102 *
2103 * POINT TO THE START OF THE SYMBOL TABLE
2104 *
2105 0DFA C020 MOV @SYMTAB,SYMPTR POINT TO SYMBOL TABLE
2106 0DFC 1154'
2107 0DFE 0208 LI NSBIT,1 MARK AS FIRST SYMBOL
2108 0E00 0001
2109 *
2110 * FIND AND PROCESS RECENTLY DEFINED SYMBOLS
2111 *
2112 SCHK
2113 0E02 06A0 BL @GTSVL GET SYMBOL VALUE
2114 0E04 0D78'
2115 0E06 C805 MOV VALUE,@FXFWP+SVAL+SVAL
2116 0E08 15DE'
2117 0E0A 06A0 BL @GTSFL GET SYMBOL FLAGS
2118 0E0C 0D52'
2119 0E0E 0282 CI FLAGS,3 IS SYMBOL RECENTLY DEFINED
2120 0E10 0003
2121 0E12 1504 JGT NX1 NO
  
```

2116	0E14	0420	BLWP	@FIXFRF	YES, FIX UP FWD REFS
	0E16	15E6			
2117	0E18	B420	AB	@BLANK, *SYMPTR	CLEAR AS REC DEF'D
	0E1A	17DC			
2118					
2119					
2120					
2121					
2122	0E1C	0588	INC	NSBIT	INCREMENT SYMBOL NUMBER
2123	0E1E	0603	DEC	SYMM	POINT TO NEW SYMBOL
2124	0E20	C003	MOV	SYMM, SYMPTR	GET NEW SYMBOL
2125	0E22	8800	C	SYMPTR, @ENDST	DONE?
	0E24	0B38			
2126	0E26	15ED	JGT	SCHK	NO
2127	0E28	04E0	CLR	@NEWFRF	YES, CLEAR RESOLVED FLAG
	0E2A	006C			
2128	0E2C	0380	RTWP		RETURN

*
 * FIND NEXT SYMBOL IN SYMBOL TABLE
 *

NX1

```

2131 * TITLE: REFDEF
2132 * OUTPUT EXTERNAL REFS AND DEFS
2133 * REVISION: 03/01/74
2134 * ORIGINAL
2135 * COMPUTER: 990, ASSEMBLY
2136 * ABSTRACT:
2137 *
2138 * SYMBOL TABLE IS EXAMINED. EXTERNALLY REF'D
2139 * OR DEF'D SYMBOLS ARE PLACED INTO THE OBJECT
2140 * CODE AND UNDEFINED SYMBOLS ARE PRINTED IN THE
2141 * UNDEFINED SYMBOLS LIST.
2142 * CALLING SEQUENCE:
2143 * BLWP @REFDEF
2144 * ADDITIONAL NOTES:
2145 * WORKSPACE = SYMBOL (SHARED WITH SENTRY,
2146 * SRCSYM, SYMCLR)
2147 * THE SYMBOL TABLE FLAG VALUE IS USED AS A JUMP
2148 * TABLE (FTAB) INDEX
2149 * ROUTINES CALLED: GTSFL, GTSVL, BINHEX, GTSNM,
2150 * CLROBJ, PRINT
2150 REFDEF
2151 0E2E 0D32' DATA SYMBOL WORKSPACE
2152 0E30 0E32' DATA AFDDBG START
2153 AFDDBG
2154 *
2155 * POINT TO THE START OF THE SYMBOL TABLE
2156 *
2157 0E32 C020 MOV @SYMTAB, SYMPTR POINT TO SYMBOL TABLE
2158 0E34 1154'
2159 0E36 0206 LI NWSCT, 5 SET SYMBOLS-ON-CARD COUNT
2160 0E38 0005
2161 0E3A 020C LI OUTS, OBJRCD POINT INTO OBJECT BUFFER
2162 0E3C 0078'
2163 NWSYM2
2164 *
2165 * WHILE SYMBL TABLE HAS AN ENTRY
2166 *
2167 0E3E 8800 C SYMPTR, @ENDST
2168 0E40 0B38'
2169 0E42 1236 JLE NXSYMO
2170 *
2171 * BRANCH ACCORDING TO FLAG
2172 *
2173 0E44 06A0 BL @GTSFL GET SYMBOL FLAGS
2174 0E46 0D52'
2175 0E48 0822 MOVB @FTAB(FLAGS), @JMPF+1
2176 0E4A 1A50'
2177 0E4C 0E4F'
2178 0E4E 10FF JMPF JMP * JUMP ON FLAG VALUE
2179 *
2180 * PROCESS EXTERNAL DEF
2181 *
2182 * EXTDEF
2183 0E50 0208 LI NSBIT, >3500 MARK AS RELOCATABLE DEF
2184 0E52 3500
  
```

2177	0E54	1002		JMP	DROUT	PLACE IN OBJECT BUFFER
2178			EXTDFA			
2179	0E56	0208		LI	NSBIT, >3600	MARK ABSOLUTE DEF
	0E58	3600				
2180			*			
2181			*		OUTPUT OBJECT	
2182			*			
2183				DROUT		
2184	0E5A	06A0		BL	@GTSVL	GET SYMBOL VALUE
	0E5C	0D78				
2185	0E5E	DF08		MOVB	NSBIT, *OUTS+	PLACE TAG IN OBJECT BUFFER
2186	0E60	C80C		MOV	OUTS, @SYMLOC	POINT INTO OBJECT BUFFER
	0E62	0E6A				
2187	0E64	0420		BLWP	@BINHEX	CONVERT TO ASCII
	0E66	0BE0				
2188	0E68	0D3C		DATA	VL	VALUE OF SYMBOL
2189	0E6A	0000	SYMLOC	DATA	0	LOCATION IN OBJECT BUFFER
2190	0E6C	022C		AI	OUTS, 4	POINT TO NAME AREA
	0E6E	0004				
2191	0E70	0207		LI	SCNT2, 6	SET CHARACTER COUNT
	0E72	0006				
2192			LPOT			
2193	0E74	DF20		MOVB	@BLANK, *OUTS+	BLANK OUT NAME AREA
	0E76	17DC				
2194	0E78	0607		DEC	SCNT2	
2195	0E7A	15FC		JGT	LPOT	
2196	0E7C	022C		AI	OUTS, -6	RESET OBJECT POINTER
	0E7E	FFFA				
2197	0E80	06A0		BL	@GTSNM	POINT TO SYMBOL NAME
	0E82	0D84				
2198	0E84	0207		LI	SCNT2, 6	SET CHARACTER COUNT
	0E86	0006				
2199			LPOT2			
2200	0E88	DF33		MOVB	*SYMMN+, *OUTS+	MOVE IN SYMBOL NAME
2201	0E8A	0607		DEC	SCNT2	
2202	0E8C	0601		DEC	COUNT	
2203	0E8E	15FC		JGT	LPOT2	
2204	0E90	A307		A	SCNT2, OUTS	ADJUST OBJECT POINTER
2205	0E92	0606		DEC	NWSCT	ROOM ON CARD?
2206	0E94	1508		JGT	NXSYM	YES - GET NEXT SYMBOL
2207	0E96	C80C		MOV	OUTS, @OTWP+OBJPT+OBJPT	NO
	0E98	16EE				
2208	0E9A	0420		BLWP	@CLR0BJ	CLEAR OBJECT BUFFER
	0E9C	17BE				
2209	0E9E	0206		LI	NWSCT, 5	RESET SYMBOL-ON-CARD COUNT
	0EA0	0005				
2210	0EA2	020C		LI	OUTS, OBJRCD	RESET OBJECT BUFFER POINTER
	0EA4	0078				
2211			*			
2212			*		EXAMINE NEXT SYMBOL IN SYMBOL TABLE	
2213			*			
2214				NXSYM		
2215	0EA6	06A0		BL	@GTSVL	POINT TO LAST WORD IN ENTRY
	0EA8	0D78				
2216	0EAA	0603		DEC	SYMMN	POINT TO NEXT SYMBOL

OUTPUT EXT REFS AND DEFS 948925-9901**

```

2217 0EAC 0003      MOV  SYMMN, SYMPTR
2218 0EAE 10C7      JMP  NWSYM2
2219                *
2220                *      END OF WHILE LOOP
2221                *
2222                NXSYMO
2223 0EB0 0286      CI   NWSCT, 5          ANY OBJECT UNPUNCHED?
      0EB2 0005
2224 0EB4 1304      JEQ  ENDIT           NO - EXIT
2225 0EB6 080C      MOV  OUTS, @OTWP+OBJPT+OBJPT  YES
      0EB8 16EE
2226 0EBA 0420      BLWP @CLROBJ        CLEAR OUT OBJECT BUFFER
      0EBC 17BE
2227                ENDIT
2228 0EBE 0380      RTWP           RETURN
2229                *
2230                *      PROCESS EXTERNAL REF
2231                *
2232                EXTRFR
2233 0EC0 0208      LI   NSBIT, >3300    MARK EXT REF IN RELOC CODE
      0EC2 3300
2234 0EC4 10CA      JMP  DROUT        PLACE IN OBJECT BUFFER
2235                EXTRFA
2236 0EC6 0208      LI   NSBIT, >3400    MARK EXT REF IN ABS CODE
      0EC8 3400
2237 0ECA 10C7      JMP  DROUT        PLACE IN OBJECT BUFFER
2238                *
2239                *      PROCESS UNDEFINED SYMBOL
2240                *
2241                UNDSM
2242 0ECC 06A0      BL   @GTSNM        GET SYMBOL NAME
      0ECE 0D84
2243 0ED0 0208      LI   NSBIT, INSBUF-1  POINT INTO PRINT LINE
      0ED2 017B
2244 0ED4 0288      MOV  NSBIT, IOPARM  SET UP I/O ROUTINE PARM
2245                UNDLF
2246 0ED6 DE33      MOVB *SYMMN+, *NSBIT+  PLACE SYMBOL NAME IN LINE
2247 0ED8 0601      DEC  COUNT
2248 0EDA 15FD      JGT  UNDLF
2249 0EDC DE20      MOVB @LF, *NSBIT+    PLACE LINE FEED IN LINE
      0EDE 0100
2250 0EE0 DE20      MOVB @CR, *NSBIT     PLACE CARRIAGE RETURN IN LINE
      0EE2 0101
2251 0EE4 0420      BLWP @PRINT        PRINT SYMBOL NAME
      0EE6 19CE
2252 0EE8 10DE      JMP  NXSYM        GET NEXT SYMBOL

```



```

2255 * TITLE: OPSRCH
2256 * SEARCH OPERATOR AND DIRECTIVE TABLE
2257 * REVISION: 03/01/74
2258 * ORIGINAL
2259 * COMPUTER: 990, ASSEMBLY
2260 * ABSTRACT:
2261 *
2262 * THE OPERATOR MNEMONIC AND DIRECTIVE TABLE IS
2263 * SEARCHED FOR THE SYMBOL NAME IN THE OPERATOR
2264 * FIELD OF A LINE. IF THE SYMBOL IS NOT FOUND,
2265 * CONTROL WILL RETURN TO THE INSTRUCTION
2266 * IMMEDIATELY FOLLOWING THE CALL. IF THE SYMBOL
2267 * IS FOUND, THIS INSTRUCTION WILL BE SKIPPED
2268 * UPON RETURN TO THE CALLING ROUTINE. IF FOUND,
2269 * THE CODE CLASS OF THE SYMBOL WILL BE DETERMINED
2270 * CALLING SEQUENCE:
2271 * BLWP @OPSRCH
2272 * NOT- FOUND-INSTRUCTION
2273 * FOUND-INSTRUCTION
2274 * ADDITIONAL NOTES:
2275 * WORKSPACE = OPSWP (UNSHARED)
2276 * AN OPERATOR MNEMONIC AND DIRECTIVE TABLE ENTRY
2277 * CONSIST OF THREE WORDS. THE FIRST TWO WORDS
2278 * CONTAIN THE SYMBOL NAME (LEFT-JUSTIFIED, ZERO-
2279 * FILLED), AND THE THIRD WORD IS DIVIDED, INTO
2280 * TWO FIELDS. THE FIRST TWELVE BITS OF THE WORD
2281 * (0-11) HOLD THE INSTRUCTION FIELD FOR A
2282 * MNEMONIC AND A BRANCH NUMBER FOR A DIRECTIVE.
2283 * THE RIGHTMOST FOUR BITS OF THE WORD SIGNIFY A
2284 * PROCESSING TYPE. THE THIRTEEN PROCESSING TYPES
2285 * ARE:
2286 *
2287 * NUM ACTION FMTS
2288 * 0 NO ACTION NOP, VII
2289 * 1 TWO GENERAL ADDRESSES I
2290 * 2 ONE GENERAL ADDRESS IV
2291 * 3 BYTE EXPRESSION II
2292 * 4 REGISTER ADDRESS, IMMEDIATE VALUE VIII
2293 * 5 REGISTER ADDRESS VI
2294 * 6 GENERAL ADDRESS, REGISTER ADDRESS III
2295 * 7 JUMP VALUES II
2296 * 8 IMMEDIATE VALUE VIII
2297 * 9 REGISTER ADDRESS, SHIFT COUNT V
2298 * A DIRECTIVE - SEPERATE PROCESSING =
2299 * B RETURN PSEUDO-OP RT
2300 * C MAP VALUE X
2301 *
2302 * THE DIRECTIVE BRANCH NUMBERS ARE IN THE LEFT
2303 * MOST BYTE AND THEIR BRANCH VALUES ARE:
2304 * 1 AORG
2305 * 2 BES
2306 * 3 BSS
2307 * 4 BYTE
2308 * 5 DATA
2309 * 6 DEF
2310 * 7 DORG
  
```

2309		*	8	DXOP	
2310		*	9	END	
2311		*	A	EQU	
2312		*	B	EVEN	
2313		*	C	IDT	
2314		*	D	LIST	
2315		*	E	PAGE	
2316		*	F	REF	
2317		*	10	RORG	
2318		*	11	TEXT	
2319		*	12	TITL	
2320		*	13	UNL	
2321		*			
2322		*			
2323		*			THE OPERATOR TABLE IS SEARCHED THROUGH THE USE
2324		*			OF A LINEAR SEARCH.
2325	0EEA	11B0		DATA FRWP	WORKSPACE
2326	0EEC	0EEE		DATA OPSBG	START
2327				OPSBG	
2328	0EEE	04CB		CLR BRANCH	CLEAR BRANCH
2329	0EF0	0207		LI OPPTR, OPTBL	POINT TO START OF TABLE
	0EF2	01FA			
2330		*			
2331		*			COMPARE SYMBOL TO TABLE ENTRY
2332		*			
2333		*			
2334	0EF4	8DE8	C	@WORD1(OPWRD), *OPPTR+	IS FIRST WORD THE SAME
	0EF6	0000			
2335	0EF8	160B	JNE	BNXOP1	NO
2336	0EFA	8DE8	C	@WORD2(OPWRD), *OPPTR+	IS WORD2 THE SAME
	0EFC	0002			
2337	0EFE	1609	JNE	BNXOP2	NO
2338	0F00	C257	MOV	*OPPTR, OPCODE	GET OPCODE
2339	0F02	C2C9	MOV	OPCODE, BRANCH	GET BRANCH TYPE
2340	0F04	0249	ANDI	OPCODE, >FFF0	SEPARATE
	0F06	FFF0			
2341	0F08	024B	ANDI	BRANCH, >F	GET BRANCH TYPE
	0F0A	000F			
2342	0F0C	05CE	INCT	14	TAKE ALTERNATE RETURN
2343	0F0E	0380	RTWP		RETURN
2344		*			
2345		*			POSITION POINTER TO NEXT OPERATOR
2346		*			
2347				BNXOP1	
2348	0F10	05C7	INCT	OPPTR	ADJUST POINTER TO
2349				BNXOP2	
2350	0F12	05C7	INCT	OPPTR	NEXT OPERATOR
2351	0F14	0287	CI	OPPTR, OPTEND	ARE WE PAST THE TABLE
	0F16	0422			
2352	0F18	12ED	JLE	BOPLOP	NO - CONTINUE SEARCH
2353	0F1A	0380	RTWP		RETURN - NOT FOUND

2356	0F1C	0000	EXWP	DATA 0	WR0	RSVAL
2357	0F1E	0000		DATA 0	WR1	CLSTYP
2358	0F20	0000		DATA 0	WR2	CHAR
2359	0F22	0000	EXWP2	DATA 0	WR3	RESLT
2360	0F24	0000		DATA 0	WR4	RESLT2
2361	0F26	0000		DATA 0	WR5	OPRES
2362	0F28	0000		DATA 0	WR6	RELVL
2363	0F2A	0000		DATA 0	WR7	RLVAL
2364	0F2C	0000		DATA 0	WR8	OP
2365	0F2E	0000		DATA 0	WR9	SP
2366	0F30	0000		DATA 0	WR10	JVAL, FL
2367	0F32	0000		DATA 0	WR11	RETURN
2368	0F34	0000		DATA 0	WR12	ZCK2
2369	0F36	0000		DATA 0	WR13	NFWD
2370	0F38	0000		DATA 0	WR14	FSTIM
2371	0F3A	0000		DATA 0	WR15	NGFL
2372	0F3C	0000		DATA 0	WR13(2)	OLD WP
2373	0F3E	0000		DATA 0	WR14(2)	OLD PC
2374	0F40	0000		DATA 0	WR15(2)	OLD SR

```

2376      * TITLE:      EXPR
2377      *              EXPRESSION EVALUATOR
2378      * REVISION:   03/01/74
2379      *              ORIGINAL RELEASE
2380      *              03/15/76
2381      *              MODIFIED TO RUN WITH PXRMT, DORG PROCESSING
2382      *              ADDED
2383      * COMPUTER:   990, ASSEMBLY
2384      * ABSTRACT:
2385      *              EXPRESSIONS ARE EVALUATED IN A LEFT-TO-RIGHT,
2386      *              NON-PRECEDENCE FORMAT. THE EVALUATION OF AN
2387      *              EXPRESSION IS HALTED UPON ONE OF THREE
2388      *              SITUATIONS: (1) THE READING OF A NON-ALPHA-
2389      *              NUMERIC CHARACTER (OTHER THAN +, -, *, /, $) (2) A
2390      *              NON-OPERATOR CHARACTER FOLLOWING AN OPERAND, OR
2391      *              (3) THE READING OF AN EXTERNALLY REFERENCED
2392      *              SYMBOL. THE EXPRESSION MAY CONTAIN ONLY ONE
2393      *              REFERENCED SYMBOL. TWO ALTERNATIVE EXITS MAY
2394      *              BE TAKEN - THE FIRST FOR EXPRESSIONS WHICH
2395      *              CONTAIN NO FORWARD REFERENCES, THE SECOND FOR
2396      *              EXPRESSIONS WHICH DO.
2397      * CALLING SEQUENCE:
2398      *              BLWP @EXPR
2399      *              NON-FORWARD-REFERENCE-INSTRUCTION
2400      *              FORWARD-REFERENCE-INSTRUCTION
2401      * ADDITIONAL NOTES:
2402      *              WORKSPACE = EXWP/EXWP2 (UNSHARED)
2403      *              ALL SIXTEEN REGISTERS ARE USED BY OVERLAPPING
2404      *              EXWP AND EXWP2. ALL ENTRANCES ARE DEFINED
2405      *              THROUGH EXWP2 TO RETAIN THE RETURN ENVIRONMENT.
2406      *              THE FIRST INSTRUCTION IS THEN A LWPI EXWP TO
2407      *              GAIN THE TREE ADDITIONAL REGISTER FOR USE. A
2408      *              LWPI EXWP2 WILL THEN PRECEDE ANY RTWP
2409      *              INSTRUCTION. THE NFWDA FLAG WILL CAUSE AN
2410      *              ERROR RETURN UPON DISCOVERY OF A FORWARD
2411      *              REFERENCE.
2412      *              ROUTINES CALLED: GETCHR, MLBFX, SCAN, SRCSYM
2413      *              GETSVL, ER7, ER2, FRENT, ER3, ER4, PRPMD, LKCHAR
2414      *              EXPR
2415      *              0F42  0F22'  DATA EXWP2          WORKSPACE
2416      *              0F44  0F46'  DATA EXBG          START
2417      *              EXBG
2418      *              0F46  02E0    LWPI EXWP          CHANGE WORKSPACE
2419      *              0F48  0F1C'
2420      *
2421      *              INITIALIZE VALUES AND FLAGS
2422      *
2423      *              0F4A  04CF    CLR  NGFL          CLEAR NEGATE FLAG
2424      *              0F4C  04E0    CLR  @FRWP+FWDSN+FWDSN CLEAR FORWARD REF SEEN FLAG
2425      *              0F4E  11B0'
2426      *              0F50  0208    LI   OP, 2          MARK AS PLUS
2427      *              0F52  0002
2428      *              0F54  0585    INC  OPRES         SET OP AS UNRESOLVED
2429      *              0F56  04CD    CLR  NFWDA        CLEAR NO FORWARD FLAG

```

```

2427 0F58 04C3 CLR RESULT CLEAR RESULT
2428 0F5A 04C4 CLR RESULT2
2429 0F5C 04C6 CLR RELVL CLEAR RELOCATABILITY VALUE
2430 0F5E 058E INC FSTIM SET FIRST TIME FLAG
2431 *
2432 * EXAMINE CLASS OF INPUT SYMBOL AND BRANCH
2433 *
2434 EXCON
2435 0F60 04C7 CLR RLVAL MARK AS ABSOLUTE
2436 0F62 0420 BLWP @GETCHR GET NEXT CHARACTER
2437 0F64 116E'
0F66 0282 CI CHAR.>20 IN RANGE?
0F68 0020
2438 0F6A 116B JLT ERRS NO - ERROR
2439 0F6C D062 MOVB @CLASS->20(CHAR),CLSTYP GET CLASS TYPE
0F6E 019A'
2440 0F70 0981 SRL CLSTYP,8
2441 0F72 D821 MOVB @JMPTB3(CLSTYP),@JMP3+1
0F74 1A38'
0F76 0F79'
2442 0F78 10FF JMP3 JMP $ BRANCH BY CLASS TYPE
2443 *
2444 * END OF EXPRESSION - EVALUATE CORRECTNESS
2445 *
2446 *
2447 0F7A 04E0 CLR @NFWDA CLEAR NO FWD REF ALLOWED FLAG
0F7C 11B8'
2448 0F7E C38E MOV FSTIM,FSTIM ANY RESULT PRESENT?
2449 0F80 1660 JNE ERRS NO. ERROR
2450 0F82 0620 DEC @SCWP+SCPT+SCPT YES. DECREMENT SCAN POINTER
0F84 0C16'
2451 0F86 C320 MOV @FRWP+FWDSP+FWDSP,ZCK2 FWD REF SEEN?
0F88 11B8'
2452 0F8A 1608 JNE GO YES
2453 0F8C C186 MOV RELVL,RELVL NO. RELOCATABILITY VALUE?
2454 0F8E 1159 JLT ERRS LESS THAN ZERO - ERROR
2455 0F90 0286 CI RELVL,1 GREATER THAN ONE?
0F92 0001
2456 0F94 1556 JGT ERRS YES - ERROR
2457 0F96 C145 MOV OPRES,OPRES NO. OPERATOR LEFT HANGING?
2458 0F98 1654 JNE ERRS YES - ERROR
2459 0F9A 1004 JMP EXPRES NO - TAKE NORMAL EXIT
2460 *
2461 * CLEAN UP FORWARD REFERENCE TABLE ENTRY
2462 *
2463 GO
2464 0F9C 0420 BLWP @MLBFX FWD RF SEEN. FIX UP MLB IN FR
0F9E 11F2'
2465 0FA0 05E0 INCT @EXWP2+28 TAKE ALTERNATE EXIT
0FA2 0F3E'
2466 EXPRES
2467 0FA4 02E0 LWPI EXWP2 ADJUST WORKSPACE
0FA6 0F22'
2468 0FA8 0380 RTWP RETURN
2469 *
    
```

```

2470                    *      PROCESS ADDITION OPERATOR
2471                    *
2472                    OPSTK1
2473    0FAA    C145            MOV    OPRES, OPRES            ALL OPERATORS RESOLVED?
2474    0FAC    1308            JEQ    STK                    YES, STACK
2475    0FAE    10D8            JMP    EXCON                NO, IGNORE '+'
2476                    *
2477                    *      PROCESS SUBTRACTION OPERATOR
2478                    *
2479                    OPSTK2
2480    0FB0    C145            MOV    OPRES, OPRES            SLL OPERATORS RESOLVED?
2481    0FB2    1305            JEQ    STK                    YES, STACK
2482    0FB4    054F            INV    NGFL                ADJUST NEGATE FLAG
2483    0FB6    10D4            JMP    EXCON                CONTINUE
2484                    *
2485                    *      PROCESS MULTIPLY OR DIVIDE OPERATOR
2486                    *
2487                    OPSTK3
2488    0FB8    058D            INC    NFWD                SET NO FWD REF FLAG
2489    0FBA    C145            MOV    OPRES, OPRES            ALL OPERATORS RESOLVED?
2490    0FBC    1642            JNE    ERRS                NO, SYNTAX ERROR
2491                    *
2492                    *      STACK OPERATOR
2493                    *
2494                    STK
2495    0FBE    0A11            SLA    CLSTYP, 1
2496    0FC0    C201            MOV    CLSTYP, OP            STACK OPERATOR
2497    0FC2    0585            INC    OPRES                SET UNRESOLVED FLAG
2498    0FC4    04CE            CLR    FSTIM                CLEAR FIRST TIME FLAG
2499    0FC6    10CC            JMP    EXCON                CONTINUE
2500                    *
2501                    *      OBTAIN OPERAND - QUOTED STRING VALUE
2502                    *
2503                    SCQT
2504    0FC8    04C5            CLR    OPRES                CLEAR OP RESOLVED FLAG
2505    0FCA    05A0            INC    @SCWP+INCSC+INCSC      SET INSIDE STRING FLAG
2506                    0FC0    0C1C ✓
2506    0FCE    0420            BLWP @SCAN                GET CHARACTER PAIR
2507                    0FD0    0C2C ✓
2507    0FD2    1000            NOP                        DUMMY EXIT FROM SCAN
2508    0FD4    0420            BLWP @GETCHR                GET NEXT CHARACTER
2509                    0FD6    116E ✓
2509    0FD8    0282            CI    CHAR, >27            IS IT A QUOTE
2510                    0FDA    0027
2510    0FDC    1632            JNE    ERRS                NO, ILLEGAL QUOTE IN EXPR
2511    0FDE    04E0            CLR    @SCWP+INCSC+INCSC      CLEAR INSIDE STRING FLAG
2512                    0FE0    0C1C ✓
2512    0FE2    C020            MOV    @SCWP+RESLT+RESLT, RVAL    GET VALUE
2513                    0FE4    0C12 ✓
2513    0FE6    06C0            SWPB RVAL                ADJUST FOR ONLY ONE CHARACTER
2514    0FE8    D000            MOVB RVAL, RVAL
2515    0FEA    1301            JEQ    SCQT2
2516    0FEC    06C0            SWPB RVAL
2517                    SCQT2
2518    0FEE    1066            JMP    VALOP2                USE AS OPERAND
    
```

```

2519 *
2520 * OBTAIN OPERAND - SYMBOL OR CONSTANT
2521 *
2522 SCAL
2523 0FF0 0620 DEC @SCWP+SCPT+SCPT DECREMENT SCAN POINTER
      0FF2 0C15
2524 0FF4 0145 MOV OPRES, OPRES ALL OPERATORS RESOLVED?
2525 0FF6 13C1 JEQ CKER YES - TWO OPERANDS IN ROW
2526 0FF8 04C5 CLR OPRES CLEAR UNRESOLVED FLAG
2527 0FFA 0420 BLWP @SCAN GET OPERAND
      0FFC 0C2C
2528 0FFE 105C JMP VALOP SCALAR - GO PROCESS
2529 1000 0420 BLWP @SRCSYM SEARCH SYMBOL TABLE
      1002 0D94
2530 1004 1000 NOP NOT IN TABLE
2531 *
2532 * SYMBOL FOUND - BRANCH ON SYMBOL FLAG TYPE
2533 *
2534 1006 02A0 MOV @SYMBOL+FLAGS+FLAGS, JVAL
      1008 0D36
2535 100A 082A MOV @JMPTB6(JVAL), @JMP6+1
      100C 1A42
      100E 1011
2536 1010 10FF JMP6 JMP * BRANCH BY FLAG TYPE
2537 *
2538 * RELOCATABLE SYMBOL
2539 *
2540 RDF
2541 1012 0587 INC RVAL MARK AS RELOCATABLE
2542 *
2543 * ABSOLUTE SYMBOL
2544 *
2545 RDF
2546 1014 0420 RLWP @GETSVL GET SYMBOL VALUE
      1016 0D6E
2547 1018 0820 MOV @SYMBOL+VALUE+VALUE, RVAL
      101A 0D3C
2548 101C 104F JMP VALOP2 PROCESS OPERAND
2549 *
2550 * EXTERNALLY DEFINED SYMBOL
2551 *
2552 ED
2553 101E 0260 MOV @SYMBOL+SYMPTR+SYMPTR, SP POINT TO SYMBOL
      1020 0D32
2554 1022 0819 MOV *SP, RVAL ADJUST FLAG VALUE
2555 1024 0240 ANDI RVAL, >FF07
      1026 FF07
2556 1028 0260 ORI RVAL, >50 MARK AS REFERENCED
      102A 0850
2557 102C 0640 MOV RVAL, *SP CONTINUE AS FWD REF
2558 *
2559 * FORWARD REFERENCED SYMBOL
2560 *
2561 FR
2562 102E 0320 MOV @NFWDA, ZCK2 IS FWD REF ALLOWED?
  
```

1030	11B8✓			
2563	1032	1602	JNE ILF	NO - ILLEGAL FWD REF
2564	1034	C34D	MOV NFWD, NFWD	YES - IS FWD REF ACCEPTABLE?
2565	1036	130A	JEQ FRGT	YES NOW - PROCESS
2566			ILF	
2567	1038	0420	BLWP @ER7	NOT NOW - ILLEGAL FWD REF
	103A	0B62✓		
2568	103C	04E0	CLR @NFWDA	CLEAR NO FWD REF ALLOWED FLAG
	103E	11B8✓		
2569	1040	1002	JMP ERVL	ERROR EXIT
2570			ERRS	
2571	1042	0420	BLWP @ER2	SYNTAX ERROR
	1044	0B4E✓		
2572			ERVL	
2573	1046	04C3	CLR RESULT	CLEAR RESULT
2574	1048	04C6	CLR RELVL	CLEAR RELOCATABILITY
2575			EXPRX2	
2576	104A	10AC	JMP EXPREX	EXIT
2577			FRGT	
2578	104C	04CE	CLR FSTIM	CLEAR FIRST TIME FLAG
2579	104E	0420	BLWP @FRENT	SET INTO FRT ENTRY
	1050	11D0✓		
2580			FREX	
2581	1052	1086	JMP EXCON	CONTINUE
2582			*	
2583			*	EXTERNALLY REFERENCED SYMBOL - MODIFY CHAIN LINKS
2584			*	
2585			ILEX	
2586	1054	0420	BLWP @ER3	ILLEGAL EXT REF
	1056	0B52✓		
2587	1058	10F6	JMP ERVL	ERROR EXIT
2588			ER	
2589	105A	C38E	MOV FSTIM, FSTIM	IS THIS FIRST OPERAND/OPERATO
2590	105C	13FB	JEQ ILEX	NO, ILLEGAL EXT REF
2591	105E	0420	BLWP @LKCHAR	GET NEXT CHARACTER
	1060	115C✓		
2592	1062	0282	CI CHAR, >20	IS IT A BLANK
	1064	0020		
2593	1066	1306	JEQ ERCT	YES, CONTINUE
2594	1068	0282	CI CHAR, >2C	NO, IS IT A COMMA
	106A	002C		
2595	106C	1303	JEQ ERCT	YES, CONTINUE
2596	106E	0282	CI CHAR, >28	NO, IS IT A LEFT PAREN
	1070	0028		
2597	1072	16F0	JNE ILEX	NO, ILLEGAL EXTERNAL REF
2598			ERCT	
2599	1074	0420	BLWP @GETSVL	GET SYMBOL VALUE
	1076	0D6E✓		
2600	1078	C0E0	MOV @SYMBOL+VALUE+VALUE, RESULT	GET PREVIOUS VALUE
	107A	0D3C✓		
2601	107C	C18A	MOV JVAL, RELVL	USE FLAG FOR RELOCATABILITY
2602	107E	0226	AI RELVL, -12	GET PROPER VALUE
	1080	FFF4		
2603	1082	C320	MOV @PPC, ZCK2	IS PRESENT PC ABS CODE ZERO
	1084	11BA✓		


```

2604 1086 1603 JNE ERINS NO
2605 1088 C320 MOV @MAINW+RELABS+RELABS, ZCK2
      108A 042E'
2606 108C 1314 JEQ EREX YES, DO NOT REPLACE
2607 ERINS
2608 108E C2E0 MOV @DORGFL, R11 IF IN DORG, DON'T CHAIN OR
      1090 1108'
2609 1092 1611 JNE EREX SET FLAGS IN SYMBOL TABLE
2610 1094 C260 MOV @SYMBOL+SYMMN+SYMMN, SP REPLACE VALUE (CHAIN
      1096 0D38'
2611 1098 C660 MOV @PPC, *SP MOVE IN PRESENT PC
      109A 11BA'
2612 109C C260 MOV @SYMBOL+SYMPTR+SYMPTR, SP
      109E 0D32'
2613 10A0 C299 MOV *SP, FL MOVE IN FLAG
2614 10A2 024A ANDI FL, >FF07 ADJUST FLAG VALUE
      10A4 FF07
2615 10A6 C320 MOV @MAINW+RELABS+RELABS, ZCK2
      10A8 042E'
2616 10AA 1302 JEQ AER 'CODE IS ABSOLUTE
2617 10AC 022A AI FL, 8 MARK RELOCATABLE
      10AE 0008
2618 AER
2619 10B0 022A AI FL, >60 MARK AS EXTERNAL REF
      10B2 0060
2620 10B4 C64A MOV FL, *SP PLACE NEW FLAG INTO ENTRY
2621 EREX
2622 10B6 10C9 JMP EXPRX2 EXIT
2623 *
2624 * CALCULATE NEW VALUE OF EXPRESSION
2625 *
2626 VALOP
2627 10B8 C020 MOV @SCWP+RESLT+RESLT, RSVAL
      10BA 0C12'
2628 VALOP2
2629 10BC 04CD CLR NFWD CLEAR NO FWD ALLOWED FLAG
2630 10BE 04CE CLR FSTIM CLEAR FIRST TIME FLAG
2631 10C0 C3CF MOV NGFL, NGFL IS NEGATE FLAG SET?
2632 10C2 1302 JEQ NONEG NO - SKIP
2633 10C4 04CF CLR NGFL YES - CLEAR NEGATE FLAG
2634 10C6 0500 NEG RSVAL NEGATE VALUE
2635 NONEG
2636 10C8 0468 B @JMPTB5-2(OP) BRANCH ON OPERATOR
      10CA 10CA'
2637 JMPTB5
2638 10CC 1003 JMP ADDUP ADDITION
2639 10CE 1008 JMP SUB SUBTRACTION
2640 10D0 100B JMP MULT MULTIPLICATION
2641 10D2 1017 JMP DIVD DIVISION
2642 *
2643 * PERFORM ADDITION
2644 *
2645 ADDUP
2646 10D4 A187 A RLVAL, RELVL CHANGE RELOCATABILITY
2647 10D6 A0C0 A RSVAL, RESLT ADD IN VALUE
    
```

```

2648 10D8 1902          JNO  TRNEX          TEST FOR OVERFLOW
2649                    TRN
2650 10DA 0420          BLWP @ER4          TRUNCATION ERROR
      10DC 0B56
2651                    TRNEX
2652 10DE 10B9          JMP  FREX          CONTINUE
2653                    *
2654                    *   PERFORM SUBTRACTION
2655                    *
2656                    SUB
2657 10E0 6187          S    RLVAL, RELVL  CHANGE RELOCATABILITY
2658 10E2 60C0          S    RSVAL, RESULT SUBTRACT VALUE
2659 10E4 19FC          JNO  TRNEX          TEST FOR OVERFLOW
2660 10E6 10F9          JMP  TRN          TRUNCATION ERROR
2661                    *
2662                    *   PERFORM MULTIPLICATION
2663                    *
2664                    MULT
2665 10E8 06A0          BL   @PRPMD        PREPARE OPERANDS
      10EA 110E
2666 10EC 38C0          MPY  RSVAL, RESULT MULTIPLY VALUE
2667 10EE C0C3          MOV  RESULT, RESULT OVERFLOW
2668 10F0 1302          JEQ  FILM          NO
2669 10F2 0420          BLWP @ER4          TRUNCATION ERROR
      10F4 0B56
2670                    FILM
2671 10F6 C0C4          MOV  RESULT2, RESULT ALLIGN RESULT
2672                    *
2673                    *   ADJUST SIGNS
2674                    *
2675                    SINFX
2676 10F8 C3CF          MOV  NGFL, NGFL    IS NEGATE FLAG SET?
2677 10FA 13F1          JEQ  TRNEX          NO - EXIT
2678 10FC 0503          NEG  RESULT        YES - NEGATE RESULT
2679 10FE 04CF          CLR  NGFL          CLEAR NEGATE FLAG
2680 1100 10EE          JMP  TRNEX          EXIT
2681                    *
2682                    *   PERFORM DIVISION
2683                    *
2684                    DIVD
2685 1102 06A0          BL   @PRPMD        PREPARE FOR DIVIDE
      1104 110E
2686 1106 C103          MOV  RESULT, RESULT2 ADJUST FOR DIVISION
2687 1108 04C3          CLR  RESULT
2688 110A 30C0          DIV  RSVAL, RESULT DIVIDE VALUE
2689 110C 10F5          JMP  SINFX        FIX SIGNS
2690                    *
2691                    *   PREPARE SIGNS IN MULTIPLY/DIVIDE OPERATIONS
2692                    *
2693                    PRPMD
2694 110E C186          MOV  RELVL, RELVL  IS RELOCATABILITY ZERO
2695 1110 1698          JNE  ERRS          NO - ERROR
2696 1112 C1C7          MOV  RLVAL, RLVAL  IS SYMBOL RELOCATABLE?
2697 1114 1696          JNE  ERRS          YES - ERROR
2698 1116 0283          CI  RESULT, -1    IS RESULT PRESENTLY NEGATIVE
    
```

1118	FFFF			
2699	111A	1502	JGT PRP2	NO - CONTINUE
2700	111C	054F	INV NGFL	YES - ADJUST NEGATE FLAG
2701	111E	0503	NEG RESLT	NEGATE RESULT
2702			PRP2	
2703	1120	0280	CI RSVAL, -1	IS OPERAND NEGATIVE
	1122	FFFF		
2704	1124	1502	JGT PRP3	NO- CONTINUE
2705	1126	054F	INV NGFL	ADJUST NEGATE FLAG
2706	1128	0500	NEG RSVAL	NEGATE OPERAND
2707			PRP3	
2708	112A	045B	RT	RETURN

2711 * TITLE: TRUNK
2712 * TRUNCATION CHECK
2713 * REVISION: 03/01/74
2714 * ORIGINAL RELEASE
2715 * COMPUTER: 990, ASSEMBLY
2716 * ABSTRACT:
2717 * TRUNK CHECKS WORD VALUES TO DETERMINE IF THE
2718 * VALUE IS BETWEEN -128 AND 127. IF NOT, A
2719 * TRUNCATION ERROR IS CREATED AND TRUNCATION
2720 * OCCURS.
2721 * CALLING SEQUENCE:
2722 * BL @TRUNK
2723 * ADDITIONAL NOTES:
2724 * ROUTINES WHICH CALL TRUNK USE ONE OF THE
2725 * FOLLOWING WORKSPACES: MAINW, OPWP/OPWP2,
2726 * OR FXWP.
2727 * TRUNK EXPECTS THE WORD VALUE IN R0 AND WILL
2728 * RETURN THE TRUNCATED VALUE IN R0. R10 WILL BE
2729 * DESTROYED. ROUTINE CALLED: ER4
2730 TRUNK
2731 112C C280 MOV R0, R10 SAVE VALUE
2732 112E 06C0 SWFB R0 CHECK FOR TRUNCATION
2733 1130 0880 SRA R0, 8 TOP BIT SIGN EXTENDED?
2734 1132 8280 C R0, R10 YES - GOOD
2735 1134 1302 JEQ ENDTRN YES - GOOD
2736 1136 0420 BLWP @ER4 TRUNCATION ERROR
1138 0B56
2737 ENDTRN
2738 113A 045B RT RETURN

2741	113C	0016'	LKWP	DATA	SCMP+SCPT+SCPT	R0	SCPTPT
2742	113E	1160'		DATA	LKC	R1	LCKRG
2743	1140	0000		DATA	0	R2	CHAR
2744	1142		INSTWD	BSS	4	R3-R4	
2745	1146	0000		DATA	0	R5	SYM
2746	1148	0009		DATA	>09	R6	TABCHR
2747	114A		SYMBL	BSS	6	R7-R9	
2748	1150	0020		DATA	>20	R10	BLNKR
2749	1152			BSS	2	R11	RETURN
2750	1154	0038'	SYMTAB	DATA	SYMT	R12	SYMBOL TABLE POINTER
2751	1156	0000		DATA	0	R13	OLD WP
2752	1158	0000		DATA	0	R14	OLD PC
2753	115A	0000		DATA	0	R15	OLD SR

```

2755            * TITLE:      LKCHAR/GETCHR
2756            *            CHARACTER SCANNER
2757            * REVISION: 03/01/74
2758            *            ORIGINAL RELEASE
2759            * COMPUTER: 990, ASSEMBLY
2760            * ABSTRACT:
2761            *            TWO ROUTINES ARE USED TO SCAN THE INPUT LINE AN
2762            *            RETURN THE NEXT CHARACTER. THE FIRST, LKCHAR,
2763            *            WILL NOT ADVANCE THE SCAN POINTER. THE SECOND,
2764            *            GETCHR, WILL ADVANCE THE SCAN POINTER.
2765            *            FURTHERMORE, IF THE CHARACTER IS BLANK, GETCHR
2766            *            WILL ADVANCE THE SCAN POINTER TO THE NEXT
2767            *            NON-BLANK CHARACTER AFTER RETURNING THE BLANK
2768            *            AS THE CHARACTER FOUND.
2769            * CALLING SEQUENCE:
2770            *            BLWP @LKCHAR
2771            *            OR
2772            *            BLWP @GETCHR
2773            * ADDITIONAL NOTES:
2774            *            WORKSPACE = LKWP (UNSHARED)
2775            *            THE CHARACTER WILL BE RETURNED RIGHT-JUSTIFIED,
2776            *            ZERO-FILLED IN WORKSPACE REG 2 OF THE CALLING
2777            *            ROUTINE. A TAB WILL BE RETURNED AS A BLANK.
2778            LKCHAR
2779    115C    113C'            DATA LKWP            WORKSPACE
2780    115E    118A'            DATA LKCALL          START
2781            LKC
2782            *
2783            *            PICK UP CHARACTER
2784            *
2785    1160    C090            MOV   *SCPTPT, CHAR          GET CHARACTER POINTER
2786    1162    D092            MOVB  *CHAR, CHAR          GET CHARACTER
2787    1164    0982            SRL   CHAR, 8            RIGHT JUSTIFY CHARACTER
2788    1166    0182            C     CHAR, TABCHR        TAB?
2789    1168    1601            JNE   EXLK            NO - EXIT
2790    116A    C08A            MOV   BLNKR, CHAR        YES - CHANGE TO BLANK
2791            EXLK
2792    116C    045B            RT                    RETURN
2793            GETCHR
2794    116E    113C'            DATA LKWP            WORKSPACE
2795    1170    1172'            DATA GTCH            START
2796            GTCH
2797    1172    0691            BL   *LCKRG            GET CHARACTER
2798    1174    C142            MOV   CHAR, SYM        SAVE CHARACTER
2799    1176    0590            INC   *SCPTPT          INCREMENT SCAN POINTER
2800            *
2801            *            SKIP OVER IMBEDDED BLANKS
2802            *
2803    1178    8282            C     CHAR, BLNKR        IS CHARACTER A BLANK
2804    117A    1608            JNE   OCHR            NO - END
2805            LPCHR
2806    117C    0691            BL   *LCKRG            YES, GET NEXT CHARACTER
2807    117E    0590            INC   *SCPTPT          INCREMENT SCAN POINTER
2808    1180    8282            C     CHAR, BLNKR        IS CHARACTER A BLANK
  
```

```
2809 1182 13FC          JEQ  LPCHR          YES. CONTINUE LOOP
2810 1184 0610          DEC  *SCPTPT        NO. DECREMENT SCAN POINTER
2811 1186 C085          MOV  SYM. CHAR      RESTORE CHARACTER
2812 1188 1001          JMP  OCHR           END
2813                    LKCALL
2814 118A 0691          BL   *LCKRG        GET CHARACTER
2815                    *
2816                    *   RETURN CHARACTER
2817                    *
2818                    OCHR
2819 118C CB42          MOV  CHAR, @4(13)   RETURN IN CALLING WR2
      118E 0004
2820 1190 0380          RTWP              RETURN
```

```
2823 *
2824 * TITLE: LIMCHK
2825 * TITLE: LIMCHK
2826 * TITLE: LIMIT CHECK
2827 * REVISION: 03/01/74
2828 * ORIGINAL RELEASE
2829 * COMPUTER: 990, ASSEMBLY
2830 * ABSTRACT:
2831 *
2832 * THE END OF THE SYMBOL TABLE IS COMPARED TO THE
2833 * END OF FORWARD REFERENCE TABLE. IF TWO TABLES
2834 * OVERLAP, THE RUN IS ABORTED AND CONTROL PASSED
2835 * TO THE 'END' PROCESSOR AS IF THERE WERE NO END
2836 * VECTOR.
2837 * CALLING SEQUENCE:
2838 * BL @LIMCHK
2839 * ADDITIONAL NOTES:
2840 * CALLED FROM ROUTINES WITH WORKSPACE SYMBOL
2841 * OR FRWP
2842 * ROUTINE CALLED: PRINT
LIMCHK
2843 1192 8820 C @ENDST,@ENDFRT COMPARE FOR OVERLAP
1194 0B38'
1196 11B6'
2844 1198 1B0A JH LIMEND NO OVERLAP
2845 119A 020A LI IOPARM,ABORT PRINT ABORT MESSAGE
119C 0102'
2846 119E 0420 BLWP @PRINT
11A0 19CE'
2847 11A2 05A0 INC @ABRTFL SET ABORT FLAG
11A4 010C'
2848 11A6 02E0 LWPI MAINW SET WORKSPACE POINTER
11A8 0428'
2849 11AA 0460 B @ERN0EV END PROCESS
11AC 08DA'
2850 LIMEND
2851 11AE 045B RT
```



```
2854
2855 *
2856 * FORWARD REFERENCE ENTRY BUILD WORKSPACE
2857 11B0 0000 FRWP DATA 0 R0 FWDSN, FRTCNT
2858 11B2 0000 DATA 0 R1 FRTPT, FRE
2859 11B4 0000 DATA 0 R2 NFRT
2860 11B6 0000 ENDFRT DATA 0 R3 MLBPT
2861 11B8 0000 NFWDA DATA 0 R4 NFA
2862 11BA 0000 PPC DATA 0 R5 PPCRG
2863 11BC 0002 TWO DATA 2 R6 FR2
2864 11BE 0000 DATA 0 R7 OPPTR
2865 11C0 0000 DATA 0 R8 OPWRD
2866 11C2 0000 DATA 0 R9 OPCODE
2867 11C4 0000 DATA 0 R10 IOPARM
2868 11C6 0000 DATA 0 R11 BRANCH
2869 11C8 0000 DORGFL DATA 0 R12 DUMMY ORG FLAG
2870 11CA 0000 DATA 0 R13 OLD WP
2871 11CC 0000 DATA 0 R14 OLD PC
2872 11CE 0000 DATA 0 R15 OLD SR
```

```

2874 * TITLE: FRENT
2875 * FOWARD REFERENCE ENTRY BUILDER
2876 * REVISION: 03/01/74
2877 * ORIGINAL RELEASE
2878 * COMPUTER: 990, ASSEMBLY
2879 * 03/15/76
2880 * MODIFIED TO RUN WITH PXRMT, DORG PROCESSING
2881 * ADDED.
2882 * ABSTRACT:
2883 * A FORWARD REFERENCE TABLE ENTRY IS BUILT. ONLY
2884 * ONE FOWARD REFERENCE IS ALLOWED PER EXPRESSION.
2885 * THIS ROUTINE PLACES THE SYMBOL'S SYMBOL TABLE
2886 * ENTRY NUMBER AND IT'S ASSOCIATED ARITHMETIC
2887 * OPERATOR INTO THE ENTRY.
2888 * CALLING SEQUENCE:
2889 * BLWP @FRENT
2890 * ADDITIONAL NOTES:
2891 * WORKSPACE = FRWP (SHARED WITH MLBFX, OUTSYM)
2892 * A FORWARD REFERENCE TABLE ENTRY CONTAINS FOUR
2893 * WORDS: LOCATION OF EXPRESSION
2894 * ENTRY CODE (ST NUMBER AND OPERATOR)
2895 * CUMULATIVE SUM OF EXPRESSION
2896 * FLAG WORD
2897 * THE LEFTMOST BYTE OF THE FLAG WORD CONTAINS THE
2898 * PRESENT ACCUMULATED RELOCATABILITY VALUE OF THE
2899 * EXPRESSION. THE RIGHTMOST BYTE CONTAINS A FLAG
2900 * CODE:
2901 * 0 - ABSOLUTE BYTE 2 - ABSOLUTE WORD
2902 * 1 - RELOCATABLE BYTE 3 - RELOCATABLE WORD
2903 * THE ABSOLUTE OR RELOCATABLE NATURE REFERRED TO
2904 * HERE IS THAT OF THE LOCATION AT WHICH THE
2905 * FORWARD REFERENCE IS MADE, AND THE WORD/BYTE
2906 * PORTION REFERS TO THE NATURE OF THE INSTRUCTION
2907 * INVOLVING THE REFERENCE. THE JUMP INSTRUCTIONS
2908 * ARE ONLY ONES FOR WHICH FORWARD REFS IN
2909 * EXPRESSIONS ARE ALLOWED TO CREATE BYTE ENTRIES.
2910 * ALL OTHER FORWARD REFERENCES MUST BE IN
2911 * EXPRESSIONS WHICH SATISFY FULL WORD ENTRIES.
2912 FRENT
2913 11D0 11B0' DATA FRWP WORKSPACE
2914 11D2 11D4' DATA FRBG START
2915 FRBG
2916 11D4 C2E0 MOV @DORGFL,R11 DO NOT BUILD FRT IF IN DORG
2917 11D6 11C8'
2918 11D8 1301 JEQ FRB
2919 11DA 0380 RTWP
2920 *
2921 * SET APPROPRIATE FLAGS FOR EXPRESSION PROCESSING
2922 *
2923 FRB
2924 11DC 0580 INC FWDSEN SET FWD SEEN FLAG
2925 11DE 0584 INC NFA SET NO FWD REF ALLOWED FLAG
2926 *
2927 * ESTABLISH ENTRY CODE WORD
  
```

```
2927          *  
2928 11E0 C060      MOV @SYMBOL+NSBIT+NSBIT,FRE  GET SYMBOL NUMBER  
      11E2 0D42  
2929 11E4 81A0      C @EXWP+OP+OP,FR2  IS OPERATOR A PLUS  
      11E6 0F2C  
2930 11E8 1302      JEQ INSTOP        YES, OP ALREADY INSERTED  
2931 11EA 0261      ORI FRE,>8000     SET NEGATIVE FLAG  
      11EC 8000  
2932          INSTOP  
2933 11EE C081      MOV FRE,NFRT      PLACE INTO STORAGE  
2934 11F0 0380      RTWP            RETURN
```

```

2937      * TITLE:      MLBFX
2938      *              FIX FOWARD REFERENCE BLOCK
2939      * REVISION:    03/01/74
2940      *              ORIGINAL RELEASE
2941      *              03/15/76
2942      *              MODIFIED TO RUN WITH PXRMR, DORG PROCESSING
2943      *              ADDED
2944      * COMPUTER:    990, ASSEMBLY
2945      * ABSTRACT:
2946      *              THIS ROUTINE COMPLETES THE BUILDING OF THE
2947      *              FORWARD REFERENCE BLOCK FOR AN EXPRESSION BY
2948      *              FILLING IN THE LOCATION WORD, THE CUMULATIVE
2949      *              SUM, AND THE FLAG WORD NECESSARY. THE FLAG WORD
2950      *              IS SET AS BEING A WORD RATHER THAN A BYTE ENTRY
2951      *              THIS WILL BE RESET AS A BYTE ONLY BY JUMP
2952      *              PROCESSORS AFTER THE EVALUATION OF THE
2953      *              EXPRESSION. THE LIMIT CHECK IS TESTED UPON
2954      *              COMPLETION.
2955      * CALLING SEQUENCE:
2956      *              BLWP @MLBFX
2957      * ADDITIONAL NOTES:
2958      *              WORKSPACE = FRWP (SHARED WITH FRENT, OUTSYM)
2959      *              ROUTINE CALLED: LIMCHK
2960      * MLBFX
2961      11F2  11B0'   DATA FRWP          WORKSPACE
2962      11F4  11F6'   DATA MLBG          START
2963      * MLBG
2964      11F6  C2E0    MOV  @DORGFL,R11          DO NOT BUILD FRT IF IN DORG
2965      11F8  11C8'
2966      11FA  1301    JEQ  MLB
2967      11FC  0380    RTWP
2968      *
2969      * COMPLETE FORWARD REFERENCE BLOCK
2970      *
2971      * MLB
2972      11FE  0583    INC  MLBPT          POINT TO NEXT WORD IN FRT
2973      1200  CCC5    MOV  PPCRG,*MLBPT+    MOVE LOCATION IN
2974      1202  CCC2    MOV  NFRT,*MLBPT+    MOVE IN FRT ENTRY
2975      1204  CCE0    MOV  @EXWP+RESLT+RESLT,*MLBPT+    CUM SUM
2976      1206  0F22'
2977      1208  DCE0    MOV  @EXWP+RELVL+RELVL+1,*MLBPT+    EXPR RELOC
2978      120A  0F29'
2979      120C  C046    MOV  FR2,FRE          SET FLAG AS WORD
2980      120E  A060    A    @MAINW+RELABS+RELABS,FRE    LOCATION RELOC
2981      1210  042E'
2982      1212  06C1    SWPB FRE          MARK AS RELOC OR ABS
2983      1214  D4C1    MOV  FRE,*MLBPT    PLACE IN ENTRY
2984      *
2985      * TEST OVERLAP WITH SYMBOL TABLE
2986      *
2987      1216  06A0    BL   @LIMCHK          TEST OVERLAP
2988      1218  1192'
2989      121A  0380    RTWP          RETURN
  
```

```

2987 *
2988 * TITLE: OUTSYM
2989 * OUTPUT UNDEFINED SYMBOLS
2990 * REVISION: 03/01/74
2991 * ORIGINAL RELEASE
2992 * 03/15/76
2993 * MODIFIED TO RUN WITH PXRMR
2994 * COMPUTER: 990. ASSEMBLY
2995 * ABSTRACT:
2996 * AN UNDEFINED SYMBOL ERROR (1) OUTPUT FOR EACH
2997 * LOCATION IN A REMAINING FORWARD REFERENCE TABLE
2998 * BLOCK.
2999 * CALLING SEQUENCE:
3000 * BLWP @OUTSYM
3001 * ADDITIONAL NOTES:
3002 * WORKSPACE = FRWP (SHARED WITH FRENT, MLBFX)
3003 * ROUTINES CALLED: BINHEX, PRINT
3004 OUTSYM
3005 121C 11B0' DATA FRWP, OUTBG WORKSPACE, START
      121E 1220'
3006 OUTBG
3007 1220 C003 MOV MLBPT, FRTCNT POINT TO END OF FRT
3008 1222 0280 CI FRTCNT, FRT-1 COMPARE TO BEGINNING OF FRT
      1224 009F
3009 1226 1316 JEQ OUTEX EQUAL=EXIT
3010 1228 0201 LI FRTPT, FRT NOT EQUAL, UNDEFINED SYMBOLS
      122A 00A0
3011 122C 020A LI IOPARM, UNDMMSG POINT TO ERROR MESSAGE
      122E 0036'
3012 1230 0420 BLWP @PRINT PRINT MESSAGE
      1232 19CE'
3013 UNDEF
3014 1234 C801 MOV FRTPT, @PTLOC3 POINT TO LOCATION ADDRESS
      1236 123C'
3015 1238 0420 BLWP @BINHEX CONVERT LOCATION ADDR
      123A 0BE0'
3016 123C 123C' PTLOC3 DATA $ TO HEX AND PLACE
3017 123E 0056' DATA UNDLOC IN MESSAGE
3018 1240 020A LI IOPARM, UNLMSG POINT TO ERROR MESSAGE
      1242 0051'
3019 1244 0420 BLWP @PRINT PRINT MESSAGE
      1246 19CE'
3020 1248 05A0 INC @NUMERR INCR NUMBER OF ERRORS
      124A 0B44'
3021 124C 0221 AI FRTPT, 8 POINT TO NEXT ENTRY
      124E 0008
3022 1250 00C1 C FRTPT, MLBPT PAST END OF FRT?
3023 1252 11F0 JLT UNDEF NO. CONTINUE
3024 OUTEX
3025 1254 0380 RTWP EXIT
  
```

3028	1256	0000	OPWP	DATA 0	WR0	EXPREG
3029	1258	116E		DATA GETCHR	WR1	CGET
3030	125A	0000		DATA 0	WR2	BVAL, TCHR
3031	125C	0000	OPWP2	DATA 0	WR3	NXTWD, NXWD
3032	125E	0000		DATA 0	WR4	STEP, INSFLG
3033	1260	0000		DATA 0	WR5	FLGREG
3034	1262	0000		DATA 0	WR6	TEMPIN
3035	1264	0000	INST	DATA 0	WR7	INSTRG
3036	1266	0000		DATA 0	WR8	THOLD, RSV3
3037	1268	0000		DATA 0	WR9	RSV
3038	126A	0000		DATA 0	WR10	SCVAL
3039	126C	0000		DATA 0	WR11	RETURN
3040	126E	0000		DATA 0	WR12	REGREG
3041	1270	1552		DATA RA	WR13	RAPT
3042	1272	0F42		DATA EXPR	WR14	EXPRT
3043	1274	002C		DATA >2C	WR15	COMMA
3044	1276	0000		DATA 0	WR13(2)	OLD WP
3045	1278	0000		DATA 0	WR14(2)	OLD PC
3046	127A	0000		DATA 0	WR15(2)	OLD SR

```

3048 * TITLE: OPTYPE
3049 * OPERATOR TYPE PROCESSORS
3050 * REVISION: 03/01/74
3051 * ORIGINAL RELEASE
3052 * COMPUTER: 990, ASSEMBLY
3053 * ABSTRACT:
3054 * THE DIFFERENT TYPES OF OPERATORS REQUIRE MANY
3055 * DIFFERENT TYPES OF PROCESSING OF THEIR OPERAND
3056 * FIELDS. A BRANCH IS TAKEN BASED ON TYPE AND THE
3057 * FINAL VALUE OF THE INSTRUCTION IS PLACED IN
3058 * THE OBJECT CODE.
3059 * CALLING SEQUENCE:
3060 * BLWP @OPTYPE
3061 * ADDITIONAL NOTES:
3062 * WORKSPACE = OPWP/OPWP2 (SHARED WITH XOPG
3063 * AND OPERAND)
3064 * ALL SIXTEEN REGISTERS ARE UTILIZED THROUGH THE
3065 * OVERLAP OF OPWP2 AND OPWP. ALL ENTRIES ARE MADE
3066 * THROUGH OPWP2 AND ARE APPROPRIATELY MODIFIED TO
3067 * USE OPWP. ALL RETURNS ARE ACCOMPLISHED THROUGH
3068 * OPWP2.
3069 * ROUTINES CALLED: SCAN, SRCSYM, GETCHR, GAD, ER4,
3070 * AGET1, EXPR, RA, ER2, BINHEX, TRUNK, PRNTLN, FWDJ,
3071 * OUTOBJ, PCPRN, ERS
3072 OPTYPE
3073 127C 125C' DATA OPWP2 WORKSPACE
3074 127E 1280' DATA OPBG START
3075 OPBG
3076 1280 02E0 LWPI OPWP ADJUST WP
3077 1282 1256'
3078 *
3079 * OBTAIN INSTRUCTION WORD AND BRANCH CODE
3080 *
3081 1284 04C4 CLR INSFLG CLEAR INSTRUCTION FLAG
3082 1286 0203 LI NXWD, INSTWD INITIALIZE NEXT WD PTR
3083 1288 1142'
3084 128A C1E0 MOV @FRWP+OPCODE+OPCODE, INSTRG GET OPCODE
3085 128C 11C2'
3086 128E C0A0 MOV @FRWP+BRANCH+BRANCH, BVAL GET BRANCH
3087 1290 11C6'
3088 1292 0A12 SLA BVAL, 1 PREPARE FOR BRANCH
3089 1294 C0A2 MOV @JMPTB7(BVAL), BVAL
3090 1296 129A'
3091 *
3092 * BRANCH ON CODE
3093 *
3094 1298 0452 B *BVAL
3095 JMPTB7
3096 129A 13F6' DATA AINST2
3097 129C 12F0' DATA TYP1
3098 129E 1316' DATA TYP2
3099 12A0 1322' DATA TYP3
3100 12A2 134A' DATA TYP4
3101 12A4 136E' DATA TYP5

```

3097	12A6	1374'		DATA TYP6	
3098	12A8	1386'		DATA TYP7	
3099	12AA	1354'		DATA TYP8	
3100	12AC	1308'		DATA TYP9	
3101	12AE	063C'		DATA DIPR	
3102	12B0	131C'		DATA TYPB	
3103	12B2	12B4'		DATA TYPC	
3104			*		
3105			*	TYPC - TERM	0 <= EXP => 7
3106			*		
3107			TYPC		
3108	12B4	0420		BLWP @SCAN	GET TERM
	12B6	0C2C'			
3109	12B8	1010		JMP VCKP	NUMERIC VALUE
3110	12BA	0420		BLWP @SRCSYM	SEARCH SYMBOL TABLE
	12BC	0D94'			
3111	12BE	1005		JMP ILFRM	NOT IN TABLE - ERROR
3112	12C0	C0A0		MOV @SYMBOL+FLAGS+FLAGS, BVAL	TEST FLAGS
	12C2	0D36'			
3113	12C4	0222		AI BVAL, -8	DEFINED?
	12C6	FFF8			
3114	12C8	1103		JLT GDTRM	YES - GOOD
3115			ILFRM		
3116	12CA	0420		BLWP @ERS	ILLEGAL FORWARD TERM
	12CC	0B66'			
3117	12CE	103C		JMP TYP0	OUTPUT OBJECT
3118			GDTRM		
3119	12D0	0420		BLWP @GETSVL	GET SYMBOL VALUE
	12D2	0D6E'			
3120	12D4	C020		MOV @SYMBOL+VALUE+VALUE, EXPREG	
	12D6	0D3C'			
3121	12D8	1002		JMP VCK	CHECK VALUE
3122			VCKP		
3123	12DA	C020		MOV @SCWP+RESLT+RESLT, EXPREG	GET SCAN VALUE
	12DC	0C12'			
3124			VCK		
3125	12DE	C280		MOV EXPREG, SCVAL	CHECK FOR TRUNCATION ERROR
3126	12E0	0240		ANDI EXPREG, 7	IS VALUE IN THREE BITS
	12E2	0007			
3127	12E4	8280		C EXPREG, SCVAL	
3128	12E6	1302		JEQ VOK	VALUE OK
3129	12E8	0420		BLWP @ER4	TRUNCATION ERROR
	12EA	0B56'			
3130			VOK		
3131	12EC	E1C0		SOC EXPREG, INSTRG	MOVE VALUE INTO INST
3132	12EE	102C		JMP TYP0	OUTPUT INSTRUCTION
3133			*		
3134			*	TYP1 - GAD, GAD	
3135			*		
3136			TYP1		
3137	12F0	06A0		BL @AGET1	MOVE FIRST GEN ADD INTO INST
	12F2	1308'			
3138	12F4	0411		BLWP *CGET	GET CHARACTER
3139	12F6	83C2		C TCHR, COMMA	COMMA?
3140	12F8	1620		JNE SER	NO - SYNTAX ERROR

3141	12FA	06A0	BL	@GAD	GET NEXT GENERAL ADDRESS
	12FC	14CC✓			
3142	12FE	0AA5	SLA	FLGREG, 10	GET TAG DEST FIELD
3143	1300	E1C5	SOC	FLGREG, INSTRG	MOVE INTO INST REG
3144	1302	0A6C	SLA	REGREG, 6	MOVE IN REGISTER VALUE
3145	1304	E1CC	SOC	REGREG, INSTRG	
3146	1306	1020	JMP	TYP0	OUTPUT INSTRUCTION
3147			*		
3148			AGET1		
3149	1308	C20B	MOV	11, RSV3	SAVE RETURN
3150	130A	06A0	BL	@GAD	GET GEN ADD
	130C	14CC✓			
3151	130E	0AA5	SLA	FLGREG, 4	MOVE TAG SOURCE INTO INST
3152	1310	E1C5	SOC	FLGREG, INSTRG	
3153	1312	E1CC	SOC	REGREG, INSTRG	MOVE REG SOURCE INTO INST
3154	1314	0458	B	*RSV3	RETURN
3155			*		
3156			*	TYP2 - GAD	
3157			*		
3158			TYP2		
3159	1316	06A0	BL	@AGET1	GET FIRST GEN ADD INTO INST
	1318	1308✓			
3160	131A	1016	JMP	TYP0	OUTPUT INSTRUCTION
3161			*		
3162			*	TYPB - COMPLETE RETURN INSTRUCTION	
3163			*		
3164			TYPB		
3165	131C	0227	AI	INSTRG, >B	COMPLETE INSTRUCTION
	131E	000B			
3166	1320	106A	JMP	AINST2	OUTPUT INSTRUCTION
3167			*		
3168			*	TYP3 - WELL - DEFINED BYTE EXPRESSION	
3169			*		
3170			TYP3		
3171	1322	05A0	INC	@NFWDA	SET NO FWD ALLOWED FLAG
	1324	11B8✓			
3172	1326	041E	BLWP	*EXPT	GET EXPRESSION
3173	1328	1000	NOP		FWD REF NOT SEEN
3174	132A	C020	MOV	@EXWP+RESLT+RESLT, EXPREG	SAVE RESULT
	132C	0F22✓			
3175	132E	06A0	BL	@TRUNK	TEST FOR TRUNCATION
	1330	112C✓			
3176	1332	0240	ANDI	EXPREG, >FF	TRUNCATE
	1334	00FF			
3177	1336	E1C0	SOC	EXPREG, INSTRG	MOVE INTO INSTRUCTION
3178	1338	1007	JMP	TYP0	OUTPUT INSTRUCTION
3179			*		
3180			SER		
3181	133A	0420	BLWP	@ER2	SYNTAX ERROR
	133C	0B4E✓			
3182	133E	04D3	CLR	*NXTWD	CLEAR NEXT WORD
3183	1340	C184	MOV	INSFLG, TEMPIN	ADJUST INSTRUCTION TYPE
3184	1342	0A16	SLA	TEMPIN, 1	
3185	1344	A106	A	TEMPIN, INSFLG	
3186	1346	0584	INC	INSFLG	

```

3187                                TYP0
3188 1348 104F                JMP  A0INST                OUTPUT INST
3189                                *
3190                                *   TYP4 - RA, IMMEDIATE VALUE
3191                                *
3192                                TYP4
3193 134A 069D                BL   *RAPT                GET REGISTER ADDRESS
3194 134C E1CC                SOC  REGREG, INSTRG     PLACE INTO INSTRUCTION
3195 134E 0411                BLWP *CGET                GET CHARACTER
3196 1350 83C2                C    TCHR, COMMA        COMMA?
3197 1352 16F3                JNE  SER                 NO - SYNTAX ERROR
3198                                *
3199                                *   TYP8 - IMMEDIATE VALUE
3200                                *
3201                                TYP8
3202 1354 05E0                INCT @PPC                INCREMENT PRESENT PC
3203 1356 11BA                BLWP *EXPRT                GET EXPRESSION
3204 135A 1003                JMP  TYP4A                FWD REF NOT SEEN
3205 135C 0204                LI   INSFLG, 3           ADJUST INSTRUCTION TYPE
3206 135E 0003
3207 1360 1043                JMP  A0INST                OUTPUT INST
3208                                TYP4A
3209 1362 C4E0                MOV  @EXWP+RESULT+RESULT, *NXTWD  PLACE RESULT IN WORD
3210 1364 0F22                INC  INSFLG                MARK AS ABSOLUTE DEF
3211 1366 0584                A    @EXWP+RELVL+RELVL, INSFLG    RELOCATABLE ADJUST
3212 1368 A120
3213 136A 0F28                JMP  A0INST                OUTPUT INSTRUCTION
3214                                *
3215                                *   TYP5 - RA
3216                                *
3217                                TYP5
3218 136E 069D                BL   *RAPT                GET REGISTER ADDRESS
3219 1370 E1CC                SOC  REGREG, INSTRG     MOVE INTO INST
3220 1372 103A                JMP  A0INST                OUTPUT INST
3221                                *
3222                                *   TYP6 - GAD, RA
3223                                *
3224                                TYP6
3225 1374 06A0                BL   @AGET1                GET GEN ADD
3226 1376 1308                BLWP *CGET                GET CHARACTER
3227 1378 0411                C    TCHR, COMMA        COMMA?
3228 137A 83C2                JNE  SERRG                NO - SYNTAX ERROR
3229 137C 162A                BL   *RAPT                GET REGISTER ADDRESS
3230 137E 069D                SLA  REGREG, 6           SHIFT AND PLACE IN INST
3231 1380 0A6C                SOC  REGREG, INSTRG
3232 1382 E1CC                JMP  A0INST                OUTPUT INST
3233 1384 1031
3234                                *
3235                                *   TYP7 - (JUMPS) BYTE EXPRESSION
3236                                *
3237                                TYP7
3238 1386 041E                BLWP *EXPRT                GET EXPRESSION
3239 1388 1012                JMP  TY72                NO FWD REF - SKIP

```

3237	138A	01A0	MOV	@FRWP+MLBPT+MLBPT,	TEMPIN	POINT TO FLAGS IN ML
	138C	11B6				
3238	138E	75A0	SB	@TWO+1,*TEMPIN		MARK AS BYTE EXPRESSION
	1390	11BD				
3239	1392	0226	AI	TEMPIN,-3		POINT TO CUMSUM IN FRT ENTRY
	1394	FFFF				
3240	1396	D587	MOVB	INSTRG,*TEMPIN		MOVE IN INSTRUCTION
3241	1398	0420	BLWP	@BINHEX		CONVERT INSTRUCTION FOR PRINT
	139A	0BE0				
3242	139C	1264	DATA	INST		INSTRUCTION
3243	139E	0138	DATA	INSVAL		PRINT LOCATION
3244	13A0	0208	LI	THOLD,>2D2D		PLACE MINUS SIGNS ACCORDINGLY
	13A2	2D2D				
3245	13A4	06A0	BL	@FWDJ		OUTPUT FORWARD JUMP
	13A6	1458				
3246				OPEXT		
3247	13A8	02E0	LNPI	OPWP2		ADJUST WP
	13AA	125C				
3248	13AC	0380	RTWP			RETURN
3249				*		
3250				TY72		
3251	13AE	C020	MOV	@EXWP+RESLT+RESLT,	EXPREG	GET RESULT
	13B0	0F22				
3252	13B2	0640	DECT	EXPREG		SUBTRACT TWO FROM VALUE
3253	13B4	6020	S	@MAINW+PC+PC,	EXPREG	SUBT PC FROM EXPR
	13B6	0430				
3254	13B8	0810	SRA	EXPREG,1		DIVIDE BY TWO
3255	13BA	06A0	BL	@TRUNK		TEST FOR TRUNCATION
	13BC	112C				
3256	13BE	0240	ANDI	EXPREG,>FF		TRUNCATE
	13C0	00FF				
3257	13C2	E1C0	SOC	EXPREG,INSTRG		PLACE INTO INSTRUCTION
3258	13C4	8820	C	@EXWP+RELVL+RELVL,	@MAINW+RELABS+RELABS	
	13C6	0F28				
	13C8	042E				
3259	13CA	130E	JEO	A0INST		RELOCATABILITY ACCEPTABLE
3260	13CC	0420	BLWP	@ER2		UNACCEPTABLE - SYNTAX ERROR
	13CE	0B4E				
3261	13D0	1012	JMP	AINST2		OUTPUT INSTRUCTION
3262				SERRG		
3263	13D2	0420	BLWP	@ER2		SYNTAX ERROR
	13D4	0B4E				
3264	13D6	100F	JMP	AINST2		OUTPUT INSTRUCTION
3265				*		
3266				*	TYP9 - RA, SHIFT VALUE	
3267				*		
3268				TYP9		
3269	13D8	069D	BL	*RAPT		GET REGISTER ADDRESS
3270	13DA	E1CC	SOC	REGREG,INSTRG		PLACE INTO INST
3271	13DC	0411	BLWP	*CGET		GET CHARACTER
3272	13DE	83C2	C	TCHR,COMMA		COMMA?
3273	13E0	16F8	JNE	SERRG		NO - SYNTAX ERROR
3274	13E2	069D	BL	*RAPT		GET SHIFT VALUE
3275	13E4	0A4C	SLA	REGREG,4		POSITION IT AND PLACE INTO IN
3276	13E6	E1CC	SOC	REGREG,INSTRG		

```

3277 *
3278 * DETERMINE THAT INSTRUCTION IS COMPLETE
3279 *
3280 A0INST
3281 13E8 0411 BLWP *CGET GET CHARACTER
3282 13EA 0282 CI CHAR. >20 BLANK?
      13EC 0020
3283 13EE 1303 JEQ AINST2 YES - CONTINUE
3284 13F0 0282 CI CHAR. EOL NO - END OF LINE?
      13F2 000D
3285 13F4 16EE JNE SERRG NO - SYNTAX ERROR
3286 AINST2
3287 13F6 0203 LI NXWD, INSTWD POINT TO FIRST EXTRA WORD
      13F8 1142
3288 13FA 06A0 BL @PCPRN PLACE PC IN OUTPUT LINE
      13FC 1490
3289 13FE 0420 BLWP @BINHEX CONVERT INSTRUCTION VALUE
      1400 0BE0
3290 1402 1264 DATA INST INSTRUCTION
3291 1404 0138 DATA INSVAL LOCATION IN OUTPUT LINE
3292 1406 0420 BLWP @PRNTLN PRINT FIRST LINE
      1408 186A
3293 140A 05E0 INCT @MAINW+PC+PC ADJUST PROGRAM COUNTER
      140C 0430
3294 140E 0420 BLWP @OUTOBJ OUTPUT OBJECT
      1410 1706
3295 *
3296 * OUTPUT INSTRUCTION USING JUMP TABLE BASED ON NUMBER
3297 * AND TYPE OF ADDITIONAL WORDS TO INSTRUCTION
3298 *
3299 1412 0205 LI FLGREG, ABSIN POINT TO ABSOLUTE PROCESS
      1414 1478
3300 1416 020C LI REGREG, RELIN RELOCATABLE PROCESS
      1418 1472
3301 141A 0200 LI EXPREG, FWDIN FORWARD REFERENCE PROCESS
      141C 1450
3302 141E 0824 MOVW @INTBL(STEP), @JMPIN+1
      1420 1A5E
      1422 1425
3303 1424 10FF JMPIN JMP # BRANCH ON INSTRUCTION TYPE
3304 IN4
3305 1426 0695 BL *FLGREG OUTPUT ABSOLUTE DATA
3306 IN1
3307 1428 0695 BL *FLGREG OUTPUT ABSOLUTE DATA
3308 IN0
3309 142A 10BE JMP OPEXT EXIT
3310 IN5
3311 142C 0695 BL *FLGREG OUTPUT ABSOLUTE DATA
3312 IN2
3313 142E 069C BL *REGREG OUTPUT RELOCATABLE DATA
3314 1430 10BB JMP OPEXT EXIT
3315 IN6
3316 1432 0695 BL *FLGREG OUTPUT ABSOLUTE DATA
3317 IN3
3318 1434 0690 BL *EXPREG OUTPUT FWD REF DATA
  
```

3319	1436	10B8	JMP	OPEXT	EXIT
3320					
3321	1438	069C	BL	*REGREG	OUTPUT RELOCATABLE DATA
3322	143A	10F6	JMP	IN1	OUTPUT ABSOLUTE DATA
3323					
3324	143C	069C	BL	*REGREG	OUTPUT RELOCATABLE DATA
3325	143E	10F7	JMP	IN2	OUTPUT RELOCATABLE DATA
3326					
3327	1440	069C	BL	*REGREG	OUTPUT RELOCATABLE DATA
3328	1442	10F8	JMP	IN3	FWD REF
3329					
3330	1444	0690	BL	*EXPREG	OUTPUT FWD REF DATA
3331	1446	10F0	JMP	IN1	OUTPUT ABSOLUTE DATA
3332					
3333	1448	0690	BL	*EXPREG	OUTPUT FWD REF DATA
3334	144A	10F1	JMP	IN2	OUTPUT RELOCATABLE DATA
3335					
3336	144C	0690	BL	*EXPREG	OUTPUT FWD REF DATA
3337	144E	10F2	JMP	IN3	FWD REF
3338					
3339	1450	0208	LI	THOLD, >2D2D	
	1452	2D2D			
3340	1454	C808	MOV	THOLD, @INSVAL	CREATE MINUS PRINT FOR FWD RE
	1456	0138			
3341					
3342	1458	C808	MOV	THOLD, @INSVAL+2	
	145A	013A			
3343	145C	05A0	INC	@OTWP+NCONS+NCONS	SET NON-CONSECUTIVE FLAG
	145E	16E6			
3344	1460	C18B	MOV	11, TEMPIN	SAVE RETURN
3345	1462	06A0	BL	@PCPRN	PLACE PC IN OUTPUT LINE
	1464	1490			
3346					
3347	1466	0420	BLWP	@PRNTLN	PRINT LINE
	1468	186A			
3348	146A	05E0	INCT	@MAINW+PC+PC	ADJUST PC
	146C	0430			
3349	146E	05C3	INCT	NXWD	POINT TO NEXT EXTRA WORD
3350	1470	0456	B	*TEMPIN	RETURN
3351					
3352	1472	D820	MOV	@OT, @RELMRK	SET RELOCATABILITY MARK
	1474	1A6B			
	1476	013C			
3353					
3354	1478	C803	MOV	NXWD, @NXWLC	POINT TO ADDITIONAL WORD
	147A	1480			
3355	147C	0420	BLWP	@BINHEX	CONVERT WORD FOR PRINTING
	147E	0BE0			
3356	1480	1480	NXWLC	DATA #	ADDITIONAL WORD VALUE
3357	1482	0138		DATA INSVAL	LOCATION IN OUTPUT LINE
3358	1484	C18B	MOV	11, TEMPIN	SAVE RETURN
3359	1486	06A0	BL	@PCPRN	PLACE PC IN OUTPUT LINE
	1488	1490			
3360	148A	0420	BLWP	@OUTOBJ	OUTPUT OBJECT
	148C	1706			

3361	148E	10EB	JMP	OTLN2	PRINT OUTPUT LINE
3362			*		
3363			*	PUT PROGRAM COUNTER OUT FOR LINE	
3364			*		
3365			PCPRN		
3366	1490	0420	BLWP	@BINHEX	CONVERT PC
	1492	0BE0			
3367	1494	0430	DATA	MAINW+PC+PC	PC VALUE
3368	1496	0132	DATA	LOCNUM	LOCATION IN OUTPUT LINE
3369	1498	045B	RT		RETURN

```

3372 * TITLE: XOPG
3373 * DEFINED OP PROCESSOR
3374 * REVISION: 03/01/74
3375 * ORIGINAL RELEASE
3376 * COMPUTER: 990, ASSEMBLY
3377 * ABSTRACT:
3378 * THIS ROUTINE PROCESSES A DEFINED OPERATOR,
3379 * CREATING THE APPROPRIATE 'XOP' INSTRUCTION IN
3380 * ITS PLACE. IT COMPLETES PROCESSING THROUGH THE
3381 * NORMAL PROCESS FOR OUTPUTTING INSTRUCTIONS AS
3382 * ESTABLISHED IN OPTYPE.
3383 * CALLING SEQUENCE:
3384 * BLWP @XOPG
3385 * ADDITIONAL NOTES:
3386 * WORKSPACE = OPWP2/OPWP (SHARED WITH OPTYPE
3387 * AND OPERAND)
3388 * ALL SIXTEEN REGISTERS ARE UTILIZED BY CREATING
3389 * THE SAME LINKAGE AS DOES OPTYPE. THIS IS DONE
3390 * SO THAT THE PROCESSING MAY EXIT THROUGH
3391 * 'ROINST' IN OPTYPE RATHER THAN HAVING TO RETURN
3392 * TO XOPG.
3393 * ROUTINES CALLED: GETSVL, GETCHR, AGET1, (SYNFRX
3394 * AND ROINST)
3395 XOPG
3396 149A 125C' DATA OPWP2 WORKSPACE
3397 149C 149E' DATA XOPBG START
3398 XOPBG
3399 149E 02E0 LMPI OPWP ADJUST WP
3400 14A0 1256'
3401 *
3402 * PROCESS DEFINED OPERATOR AS XOP
3403 *
3404 14A2 0207 LI INSTRG.>2000 MARK AS XOP
3405 14A4 2000
3406 14A6 0420 BLWP @GETSVL GET SYMBOL VALUE
3407 14A8 0D6E'
3408 14AA C020 MOV @SYMBOL+VALUE+VALUE, EXPREG
3409 14AC 0D3C'
3410 14AE 0A60 SLA EXPREG, 6 SHIFT FOR PLACEMENT
3411 14B0 E1C0 SOC EXPREG, INSTRG PLACE INTO INSTRUCTION
3412 14B2 04C4 CLR INSEFLG CLEAR INSTRUCTION TYPE
3413 14B4 0203 LI NXWD, INSTWD POINT TO ADDITIONAL WORD
3414 14B6 1142'
3415 14B8 0411 BLWP *CGET GET CHARACTER
3416 14BA 0282 CI CHAR.>20 BLANK?
3417 14BC 0020
3418 14BE 1302 JED XOP2 YES - GOOD
3419 14C0 0460 B @SYNFRX NO - SYNTAX ERROR
3420 14C2 05FC'
3421 XOP2
3422 14C4 06A0 BL @AGET1 GET FIRST OPERAND
3423 14C6 1308'
3424 *
3425 * OUTPUT INSTRUCTION
    
```

3418

*

3419

14C8 0460

B

@AOINST

OUTPUT INSTRUCTION

14CA 13E8


```

3422 * TITLE: GAD/RA
3423 * OPERAND PROCESSORS
3424 * REVISION: 03/01/74
3425 * ORIGINAL RELEASE
3426 * COMPUTER: 990, ASSEMBLY
3427 * ABSTRACT:
3428 * TWO ROUTINES ARE USED IN THE PROCESSING OF
3429 * OPERAND FIELDS IN AN INSTRUCTION. BASED ON THE
3430 * NATURE OF THE OPERATOR, EITHER A GENERAL
3431 * ADDRESS (ONE OF FIVE ADDRESSING MODES) OR
3432 * A WORKSPACE REGISTER ADDRESS PROCESSOR IS
3433 * CALLED FOR. GAD WILL INTERPRET THE OPERAND TO
3434 * DETERMINE WHICH MODE IS TO BE USED OR RA
3435 * WILL PROCESS A WORKSPACE REGISTER ADDRESS.
3436 * APPROPRIATE PROCESSING VALUES WILL BE RETURNED
3437 * IN THE CALLING ROUTINES.
3438 * CALLING SEQUENCE:
3439 * BL @GAD OR BL @RA
3440 * ADDITIONAL NOTES:
3441 * WORKSPACE = OPWP2/OPWP (SHARED WITH OPTYPE
3442 * AND XOPG)
3443 * GAD WILL POSSIBLY CALL RA, THUS, ONE LEVEL OF
3444 * LINK RETURNS MUST BE SAVED IN WORKSPACE
3445 * REGISTER RSV.
3446 * ROUTINES CALLED: LKCHR, GETCHR, EXPR, RA, ER2,
3447 * SCAN, SRCSYM, GETSVL, ER8.
3448 GAD
3449 14CC C24B MOV 11, RSV SAVE RETURN
3450 *
3451 * INTERPRET GENERAL ADDRESS
3452 *
3453 *****
3454 14CE 04C5 CLR FLGREG CLEAR TAG
3455 14D0 04CC CLR REGREG CLEAR REGISTER VALUE
3456 14D2 0420 BLWP @LKCHAR LOOK AT CHARACTER
3457 14D4 115C CI TCHR, ATSGN IS FIRST SYMBOL '@'
3458 14D8 0040 JNE REG NO. REGISTER TYPE
3459 *
3460 * SYMBOLIC MEMORY ADDRESS AND INDEXED MEMORY ADDRESS
3461 *
3462 14DC 0411 BLWP *CGET GET CHARACTER
3463 14DE 05E0 INCT @PPC MARK PRESENT LOC
3464 14E0 11BA MOV INSFLG, TEMPIN COMPUTE NEW FLAG
3465 14E4 1302 JEQ EXPAD IF ZERO, FORGET SHIFT
3466 14E6 0A16 SLA TEMPIN, 1 MULTIPLY BY THREE
3467 14E8 A106 A TEMPIN, INSFLG
3468 EXPAD
3469 14EA 041E BLWP *EXPRT FWD REF IN EXPRESSION?
3470 14EC 1003 JMP ADDRD NO. CHECK IF ABS OR REL DEF
3471 14EE 0224 AI INSFLG, 3 ADJUST INSTRUCTION TYPE
3472 14F0 0003
  
```

3472	14F2	1003		JMP	MOVWD	MOVE WORD
3473			ADORRD			
3474	14F4	0584		INC	INSFLG	MARK AS ABSOLUTE
3475	14F6	A120		A	@EXWP+RELV+RELV, INSFLG	ADJUST IF RELOC
	14F8	0F28				
3476			MOVWD			
3477	14FA	CCE0		MOV	@EXWP+RESLT+RESLT, *NXWD+	PLACE RESULT IN WORD
	14FC	0F22				
3478	14FE	0420		BLWP	@LKCHAR	LOOK AT NEXT CHARACTER
	1500	1150				
3479	1502	0282		CI	TCHR, >20	BLANK?
	1504	0020				
3480	1506	1311		JEQ	EXIT2	YES - SYMBOLIC MEMORY EXIT
3481	1508	83C2		C	TCHR, COMMA	COMMA?
3482	150A	130F		JEQ	EXIT2	YES - SYMBOLIC MEMORY EXIT
3483	150C	0282		CI	TCHR, >28	LEFT PARENTHESIS?
	150E	0028				
3484	1510	1607		JNE	EXT2	NO - EXIT WITH ERROR
3485	1512	0411		BLWP	*CGET	GET CHARACTER
3486	1514	069D		BL	*RAPT	YES - GET REGISTER INDEX
3487	1516	05C5		INCT	FLGREG	MARK TAG AS 2
3488	1518	0411		BLWP	*CGET	GET RIGHT PAREN
3489	151A	0282		CI	CHAR, >29	IS IT RIGHT PARENTHESIS?
	151C	0029				
3490	151E	1304		JEQ	EXT3	YES - INDEXED EXIT
3491			EXT2			
3492	1520	0420		BLWP	@ER2	SYNTAX ERROR
	1522	0B4E				
3493	1524	0620		DEC	@SCWP+SCPT+SCPT	DECREMENT SCAN POINTER
	1526	0C16				
3494			EXT3			
3495	1528	0459		B	*RSV	RETURN
3496			*			
3497			EXIT2			
3498	152A	05C5		INCT	FLGREG	MARK TAG AS 2
3499	152C	0459		B	*RSV	RETURN
3500			*			
3501			*			
3502			*		INDIRECT WORKSPACE REGISTER ADDRESS,	
3503			*		WORKSPACE REGISTER ADDRESS AND	
3504			*		INDIRECT AUTO-INCREMENT	
3505			*			
3506			REG			
3507	152E	0282		CI	TCHR, ASTR	IS FIRST SYMBOL AN ASTERISK
	1530	002A				
3508	1532	1302		JEQ	INDRCT	YES, INDIRECT
3509	1534	069D		BL	*RAPT	NO - GET REGISTER ADDRESS
3510	1536	0459		B	*RSV	RETURN
3511			INDRCT			
3512	1538	0411		BLWP	*CGET	SKIP '*'
3513	153A	069D		BL	*RAPT	GET REGISTER ADDRESS
3514	153C	0420		BLWP	@LKCHAR	LOOK AT CHARACTER
	153E	1150				
3515	1540	0282		CI	TCHR, >2B	IS IT A '+'
	1542	002B				

3516	1544	1302	JEQ	EXIT4	YES, AUTOINCREMENT
3517	1546	0585	INC	FLGREG	MARK TAG AS 1
3518	1548	0459	B	*RSV	RETURN
3519			*		
3520				EXIT4	
3521	154A	0411	BLWP	*CGET	GET CHARACTER
3522	154C	0205	LI	FLGREG, 3	MARK TAG AS 3
	154E	0003			
3523	1550	0459	B	*RSV	RETURN
3524			*		
3525			*	WORKSPACE REGISTER ADDRESS PROCESSOR	
3526			*		
3527				RA	
3528	1552	0420	BLWP	@SCAN	GET SYMBOL VALUE
	1554	0C2C			
3529	1556	1010	JMP	VALCKP	NUMERICAL VALUE
3530	1558	0420	BLWP	@SRCSYM	LOOK UP SYMBOL
	155A	0D94			
3531	155C	1014	JMP	ERREX	NOT THERE - ERROR
3532	155E	C2A0	MOV	@SYMBOL+FLAGS+FLAGS, SCVAL	GET FLAGS
	1560	0D36			
3533	1562	028A	CI	SCVAL, 7	IS SYMBOL DEFINED
	1564	0007			
3534	1566	150F	JGT	ERREX	NO, ILLEGAL REGISTER
3535	1568	024A	ANDI	SCVAL, 2	IS SYMBOL ABSOLUTE
	156A	0002			
3536	156C	130C	JEQ	ERREX	NO-ILLEGAL REGISTER
3537	156E	0420	BLWP	@GETSVL	GET SYMBOL VALUE
	1570	0D6E			
3538	1572	C2A0	MOV	@SYMBOL+VALUE+VALUE, SCVAL	
	1574	0D3C			
3539	1576	1002	JMP	VALCHK	CHECK VALUE FOR RANGE
3540				VALCKP	
3541	1578	C2A0	MOV	@SCWP+RESLT+RESLT, SCVAL	
	157A	0C12			
3542				VALCHK	
3543	157C	C30A	MOV	SCVAL, REGREG	IS VALUE LESS THAN ZERO
3544	157E	1103	JLT	ERREX	YES, ERROR
3545	1580	028A	CI	SCVAL, SIXTN	IS VALUE LESS THAN SIXTEEN
	1582	0010			
3546	1584	1103	JLT	VALOUT	YES - GOOD
3547				ERREX	
3548	1586	0420	BLWP	@ERS	ILLEGAL REGISTER
	1588	0B66			
3549	158A	040C	CLR	REGREG	CLEAR TERM
3550				VALOUT	
3551	158C	045B	RT		RETURN

```
3554 *  
3555 * PRINT PAGE HEADER WORKSPACE AREA  
3556 *  
3557 PGWP  
3558 158E 0BF4' DATA BINDEC R0 DECPT  
3559 1590 0000 LNCNT DATA 0 R1 LNCTR  
3560 1592 0000 NPAGE DATA 0 R2 PGCNT  
3561 1594 00C2' DATA TITLE R3 TITPT  
3562 1596 0BC0' DATA BINVL R4 BINVP  
3563 1598 0007 TABS DATA 7,12,30,32,0 R5-R9 TAB SETTINGS BASED ON 0-59  
159A 000C  
159C 001E  
159E 0020  
15A0 0000  
3564 15A2 0000 DATA 0 R10 IOPARM  
3565 15A4 2A2A FNDMK DATA '***' R11  
3566 15A6 0000 LABSN DATA 0 R12  
3567 15A8 0000 DATA 0 R13 OLD WP  
3568 15AA 0000 DATA 0 R14 OLD PC  
3569 15AC 0000 DATA 0 R15 OLD SR
```

```

3571                    * TITLE:      PGHDR
3572                    *              PRINT PAGE HEADER
3573                    * REVISION    03/01/74
3574                    *              ORIGINAL
3575                    *              03/15/76
3576                    *              MODIFIED TO RUN WITH PXRMR
3577                    * COMPUTER: 990. ASSEMBLY
3578                    * ABSTRACT: PGHDR PRINTS A PAGE HEADING ON A NEW PAGE
3579                    *              AT LEAST EVERY FIFTY-FIVE LINES. THE LINE
3580                    *              COUNT IS THEN RESET TO FIFTY-FIVE FOR THE
3581                    *              NEXT LINE.
3582                    * CALLING SEQUENCE:
3583                    *              BLWP @PGHDR
3584                    * ADDITIONAL NOTES:
3585                    *              WORKSPACE = PGWP (UNSHARED)
3586                    *              ROUTINES CALLED: BINDEC,PRINTP
3587                    *
3588                    PGHDR
3589    15AE    158E'                    DATA PGWP,PGBG                    WORKSPACE, START
          15B0    15B2'
3590                    PGBG
3591    15B2    0283                    NOV    TITPT,IOPARM                    POINT TO TITLE
3592                    *
3593                    *    CHANGE PAGE COUNT
3594                    *
3595    15B4    0582                    INC    PGCNT                    INCREMENT PAGE COUNT
3596    15B6    0502                    NOV    PGCNT,*BINVP                    CONVERT PAGE COUNT
3597    15B8    0410                    BLWP *DECPT                    TO DECIMAL AND
3598    15BA    00FB'                    DATA PAGNUM                    PLACE IN TITLE LINE
3599                    *
3600                    *    PRINT TITLE
3601                    *
3602    15BC    0420                    BLWP @PRINTP                    PRINT TITLE
          15BE    1902'
3603                    *
3604                    *    CORRECT LINE COUNT
3605                    *
3606    15C0    0201                    LI    LNCTR,55                    SET LINE COUNT
          15C2    0037
3607    15C4    0380                    RTWP                    RETURN
  
```

			FXFWF		
3610					
3611	1506	0000	DATA 0	WR0	TEMP
3612	1508	0000	DATA 0	WR1	PTR1
3613	150A	0000	DATA 0	WR2	SABS. PTR2
3614	150C	0000	DATA 0	WR3	FRTVAL
3615	150E	0000	DATA 0	WR4	RELREG
3616	1500	0000	DATA 0	WR5	FLGREG
3617	1502	0000	DATA 0	WR6	HPTR1
3618	1504	0000	DATA 0	WR7	HPTR1V
3619	1506	11BA	DATA PPC	WR8	PPCPT2
3620	1508	11B6	DATA ENDFRT	WR9	FRTEND
3621	150A	0000	DATA 0	WR10	TEMP2
3622	150C	0000	DATA 0	WR11	RETURN
3623	150E	0000	DATA 0	WR12	SVAL
3624	15E0	0000	DATA 0	WR13	OLD WP
3625	15E2	0000	DATA 0	WR14	OLD PC
3626	15E4	0000	DATA 0	WR15	OLD SR

```

3628 * TITLE: FIXFRF
3629 * FIX UP FORWARD REFERENCE
3630 * REVISION: 03/01/74
3631 * ORIGINAL RELEASE
3632 * 03/15/76
3633 * MODIFIED TO RUN WITH PXRMTX, AND FOR DORG
3634 * PROCESSING.
3635 * COMPUTER: 990, ASSEMBLY
3636 * ABSTRACT:
3637 * FOR EACH SYMBOL THAT HAS BEEN RECENTLY DEFINED-
3638 * PREVIOUSLY REFERENCED BUT WHOSE REFERENCES HAVE
3639 * NOT YET BEEN RESOLVED-, THE FORWARD REFERENCE
3640 * TABLE ENTRIES CORRESPONDING TO THE SYMBOL ARE
3641 * RESOLVED AND THE APPROPRIATE OBJECT RECORDS ARE
3642 * OUTPUT. DIFFERENT PROCESSING OCCURS FOR
3643 * SIXTEEN-BIT AND EIGHT-BIT REFERENCES. THE
3644 * TABLE IS CONDENSED FOR EACH RESOLUTION.
3645 * CALLING SEQUENCE:
3646 * BLWP @FIXFRF
3647 * ADDITIONAL NOTES:
3648 * WORKSPACE = FXWP (UNSHARED)
3649 * ROUTINES CALLED: ER4, TRUNK, BINHEX, OUTOB, PNTLN
3650 FIXFRF
3651 15E6 1506' DATA FXFWP WORKSPACE
3652 15E8 15EA' DATA FXFBG START
3653 FXFBG
3654 15EA 0201 LI PTR1,FRT POINT TO ENTRY BEING CHECKED
3655 15EC 00A0
3656 15EE C631 MOV *PTR1+,*PFCPT2 PLACE LOCATION INTO PRESENT P
3657 15F0 C081 MOV PTR1, PTR2 POINT TO MOVING LOCATION
3658 *
3659 * EXAMINE FRT ENTRY FOR CORRESPONDENCE TO SYMBOL
3660 *
3661 CKENT
3662 15F2 8641 C PTR1,*FRTEND PAST THE END OF TABLE
3663 15F4 150C JGT EXE YES - EXIT
3664 15F6 C0D1 MOV *PTR1,FRTVAL NO. COMPARE PTR OF PTR1 TO
3665 15F8 0243 ANDI FRTVAL,>3FFF SYMBOL NUMBER
3666 15FA 3FFF
3667 15FC 80E0 C @SYMBOL+NSBIT+NSBIT,FRTVAL
3668 15FE 0D42'
3669 1600 130E JEQ FOUND2 SAME
3670 1602 CCB1 MOV *PTR1+,*PTR2+ COPY ENTRY
3671 1604 CCB1 MOV *PTR1+,*PTR2+ COPY CUNSUM
3672 1606 CCB1 MOV *PTR1+,*PTR2+ COPY RELOC AND FLAGS
3673 1608 C611 MOV *PTR1,*PFCPT2 PLACE LOCATION INTO PRESENT P
3674 160A CCB1 MOV *PTR1+,*PTR2+ COPY LOC OF NEXT NLB
3675 160C 10F2 JMP CKENT GET NEXT ENTRY
3676 *
3677 * MODIFY END OF FRT TABLE AFTER ALL CONDENSATION DONE
3678 *
3679 EXE
3680 160E 0642 DECT PTR2 POINT TO END OF FRT
3681 1610 C042 MOV PTR2, PTR1 SAVE END OF FRT
  
```

```

3679
3680 *
3681 1612 04F2 CLR *PTR2+ CLEAR FRT AREA
3682 1614 8642 C PTR2, *FRTEND PAST TABLE
3683 1616 12FD JLE CLRFRT NO - CONTINUE
3684 1618 0601 DEC PTR1 YES:
3685 161A C641 MOV PTR1, *FRTEND ADJUST END OF TABLE
3686 161C 0380 RTWP RETURN
3687
3688 *
3689 * PROCESS FOUND ENTRY
3690 *
3691 161E 0204 FOUND2
3692 1620 0100 LI RELREG, >100 MARK SYMBOL 'PLUS RELOC'
3693 1622 C0F1 MOV *PTR1+, FRTVAL TEST ADD/SUBTRACT BIT
3694 1624 1503 JGT POS MINUS FLAG NOT SET - SKIP
3695 1626 0204 LI RELREG, >FF00 MARK SYMBOL 'MINUS RELOC'
3696 1628 FF00
3697 162A 050C NEG SVAL NEGATE SYMBOL VALUE
3698 162C 01A0 POS
3699 162E 0D36 MOV @SYMBOL+FLAGS+FLAGS, HPTR1 GET SYMBOL'S FLAGS
3700 1630 0246 ANDI HPTR1, SABS ABSOLUTE?
3701 1632 0002
3702 1634 1301 JEQ RELOC SYMBOL IS RELOCATABLE
3703 1636 04C4 CLR RELREG NO. MARK SYMBOL 'ABSOLUTE'
3704 1638 C181 RELOC
3705 163A 0506 MOV PTR1, HPTR1 CREATE TEMPORARY HOLD FOR PTR
3706 163C C1D6 INCT HPTR1 POINT TO FLAGS/RELOC
3707 163E 25E0 MOV *HPTR1, HPTR1V GET VALUE
3708 1640 11BC C2C @TWO, HPTR1V IS THIS A BYTE INSTRUCTION
3709 1642 1306 JEQ BYTFIX YES - PROCESS
3710 1644 0646 DECT HPTR1 NO. CHANGE VALUE
3711 1646 AD8C A SVAL, *HPTR1+ ADD VALUE TO CONSUM
3712 1648 1913 JNO NOVFL NO OVERFLOW
3713 164A 0420 BLWP @ER4 TRUNCATION ERROR
3714 164C 0B56
3715 164E 1010 JMP NOVFL
3716
3717 *
3718 * PROCESS BYTE REFERENCE
3719 *
3720 BYTFIX
3721 1650 0606 DEC HPTR1 POINT TO LOW ORDER BYTE OF IN
3722 1652 D016 MOVB *HPTR1, TEMP GET LOW ORDER BYTE
3723 1654 0880 SRA TEMP, 8 EXTEND THE SIGN
3724 1656 A00C A SVAL, TEMP ADD IN VALUE
3725 1658 0640 DECT TEMP SUBTRACT TWO
3726 165A 0642 DECT PTR2 SUBTRACT LOCATION
3727 165C 6032 S *PTR2+, TEMP
3728 165E 0810 SRA TEMP, 1 DIVIDE BY TWO
3729 1660 0680 RL @TRUNK TEST FOR TRUNCATION
3730 1662 117C
3731 1664 0A80 SLA TEMP, 8 GET DISPLACEMENT
3732 1666 D080 MOVB TEMP, *HPTR1+ INSERT DISP

```


3727	1668	0247	ANDI HPTR1V, 1	GET RELOC OF LOCATION
	166A	0001		
3728	166C	0A87	SLA HPTR1V, 8	
3729	166E	7587	SB HPTR1V, *HPTR1	CORRECT FOR MINUS LOC ON JMP

```

3731 *
3732 * CHECK FOR PROPER RELOCATABILITY OF RESULT
3733 *
3734 NOVFL
3735 1670 B084 AB RELREG, *HPTR1+ FIX RELOC FOR RESOLVED SYMBOL
3736 1672 C156 MOV *HPTR1, FLGREG GET FLAGS VALUE
3737 1674 0606 DEC HPTR1
3738 1676 D136 MOVB *HPTR1+, RELREG GET RELOC VALUE
3739 1678 130A JEQ SETOUT CUMSUM ABSOLUTE
3740 167A 0884 SRA RELREG, 8 TEST RELOCATABILITY
3741 167C 0284 CI RELREG, 1 IS CUMSUM ACCEPTABLE RELOC
    167E 0001
3742 1680 1604 JNE TRERR NO - TRUNCATION ERROR
3743 *
3744 1682 2560 CZC @TWO, FLGREG IS THIS A BYTE INSTRUCTION
    1684 11BC
3745 1686 1603 JNE SETOUT NO - OUTPUT INSTRUCTION
3746 1688 04C4 CLR RELREG YES - CLEAR RELOCATABILITY
3747 *
3748 TRERR
3749 168A 0420 BLWP @ER4 TRUNCATION ERROR
    168C 0B56
3750 *
3751 * OUTPUT INSTRUCTION
3752 *
3753 SETOUT
3754 168E 0642 DECT PTR2 REMOVE LOCATION FROM FRT
3755 1690 C802 MOV PTR2, @PTLOC POINT TO LOCATION
    1692 1698
3756 1694 0420 BLWP @BINHEX GET ASCII REPRESENTATION
    1696 0BE0
3757 1698 1698 PTLOC DATA $ LOCATION VALUE
3758 169A 0132 DATA LOCNUM LOCATION IN OUTPUT LINE
3759 169C C801 MOV PTR1, @PTLOC2 POINT TO VALUE
    169E 16A6
3760 16A0 05C1 INCT PTR1 POINT TO FLAGS
3761 16A2 0420 BLWP @BINHEX CONVERT
    16A4 0BE0
3762 16A6 16A6 PTLOC2 DATA $ DATA
3763 16A8 0138 DATA INSVL LOCATION FOR DATA
3764 16AA D820 MOVB @BLANK, @RELMRK CLEAR RELOCATABILITY MARK
    16AC 17DC
    16AE 013C
3765 16B0 C104 MOV RELREG, RELREG IS VALUE RELOCATABLE
3766 16B2 1303 JEQ OL2 NO
3767 16B4 D820 MOVB @OT, @RELMRK YES, MARK ON OUTPUT
    16B6 1A6B
    16B8 013C
3768 OL2
3769 16BA 0245 ANDI FLGREG, 1 GET RELOCATABILITY
    16BC 0001
3770 16BE 0207 LI HPTR1V, MAINW+RELABS+RELABS
    16C0 042E
3771 16C2 C197 MOV *HPTR1V, HPTR1 DISPLACE PRESENT RELOC VALUE
  
```

3772	16C4	C5C5	MOV	FLGREG, *HPTR1V	
3773	16C6	05A0	INC	@OTWP+NCONS+NCONS	SET NON-CONSECUTIVE FLAG
	16C8	16E6'			
3774	16CA	0420	BLWP	@OUTOB	PLACE IN OBJECT BUFFER
	16CC	170A'			
3775	16CE	05A0	INC	@OTWP+NCONS+NCONS	SET NON-CONSECUTIVE FLAG
	16D0	16E6'			
3776	16D2	C5C6	MOV	HPTR1, *HPTR1V	REPLACE PRESENT RELOC VALUE
3777	16D4	C820	MOV	@FWDK, @LOCNUM+4	PLACE '**' INTO OUTPUT LINE
	16D6	15A4'			
	16D8	0136'			
3778	16DA	0420	BLWP	@PRNTLN	PRINT OUTPUT LINE
	16DC	186A'			
3779	16DE	05C1	INCT	PTR1	POINT TO LOCATION FOR NEXT ML
3780					
3781			*		
3782			*	SKIP OVER ENTRY TO CONDENSE TABLE	
			*		
3783	16E0	C611	MOV	*PTR1, *PPCPT2	ADJUST PRESENT PC
3784	16E2	CCB1	MOV	*PTR1+, *PTR2+	COPY NEW LOC
3785	16E4	1086	JMP	CKENT	CHECK NEXT ENTRY

```
3788 *  
3789 * OUTPUT OBJECT RECORD WORKSPACE AREA  
3790 *  
3791 16E6 0000 OTWP DATA 0 R0 NCONS  
3792 16E8 0000 CKSM DATA 0 R1 CKSUM  
3793 16EA 3700 DATA >3700 R2 SVN  
3794 16EC 3900 DATA >3900 R3 ABVL  
3795 16EE 0000 DATA 0 R4 OBJPT  
3796 16F0 00AF DATA OBJRCD+55 R5 CDALIM  
3797 16F2 4600 DATA >4600 R6 FVL  
3798 16F4 0000 DATA 0 R7 INSPT, HVAL  
3799 16F6 4200 DATA >4200 R8 ABDVL  
3800 16F8 00B4 DATA OBJRCD+60 R9 CDLIM  
3801 16FA 0000 DATA 0 R10 IOPARM  
3802 16FC 0000 DATA 0 R11 RETURN  
3803 16FE 0000 DATA 0 R12 CKPT  
3804 1700 0000 DATA 0 R13 OLD WP  
3805 1702 0000 DATA 0 R14 OLD PC  
3806 1704 0000 DATA 0 R15 OLD SR
```

```

3808 *
3809 * TITLE:      OUTOBJ
3810 *           OUTPUT OBJECT RECORD
3811 * REVISION:   03/01/74
3812 *           ORIGINAL
3813 *           03/15/76
3814 *           MODIFIED TO RUN WITH PXRMTX, DORG PROCESSING
3815 *           ADDED
3816 * ABSTRACT:  THIS ROUTINE PLACES OBJECT CODE INTO
3817 *           THE OBJECT BUFFER FOR OUTPUT.  IF NON-
3818 *           CONSECUTIVE CODE IS BEING ENTERED, THE
3819 *           DATA ADDRESS WILL BE PLACED INTO THE
3820 *           CODE.  OTHERWISE, JUST THE DATA WILL
3821 *           BE PUT INTO THE LINE.  WHEN THE PRESENT
3822 *           LINE WILL NOT HOLD THE FULL DATA WORD
3823 *           (AND ADDRESS, IF APPLICABLE), THE LINE
3824 *           OF OBJECT WILL BE WRITTEN BEFORE THE
3825 *           NEW OBJECT IS ADDED.
3826 *           CLR OBJ IS A SEPARATE ENTRY TO FORCE A
3827 *           BUFFER THAT IS NOT FULL TO BE WRITTEN
3828 *           TO TAPE.
3829 * CALLING SEQUENCE:
3830 *           BLWP @OUTOBJ
3831 *           OR
3832 *           BLWP @CLROBJ
3833 *           OR
3834 *           BLWP @OUTOB
3835 * ADDITIONAL NOTES:
3836 *           WORKSPACE = OTWP (UNSHARED)
3837 *           ROUTINES CALLED: OTRCD, BINHEX, WRITE
3838 *
3839 OUTOBJ
3840 1706 16E6' DATA OTWP, OTBG          WORKSPACE, START
3841 1708 170E'
3842 OUTOB
3843 170A 16E6' DATA OTWP, OUTO         WORKSPACE, START
3844 170C 1715'
3845 OTBG
3846 170E C2E0 MOV @DORGFL, R11          EXIT IF IN DORG
3847 1710 11C8'
3848 OUTO
3849 1712 1301 JEQ OUTO
3850 1714 0380 RTWP
3851 OUTO
3852 1716 04E0 CLR @OTBWP+BYTFL+BYTFL  CLEAR BYTE BOUNDARY FLAG
3853 1718 17D8'
3854 *
3855 * TEST FOR NECESSITY OF ADDRESS IN OBJECT LINE
3856 *
3857 171A C000 MOV NCONS, NCONS          CONSECUTIVE VALUE?
3858 171C 1312 JEQ OVAL              YES
3859 171E 04C0 CLR NCONS              NO, OUTPUT ADDRESS
3860 1720 8144 C OBJPT, COALIM        IS THERE ROOM?
3861 1722 1102 JLT ROOM              YES, PLACE IN LINE
3862 OUTRCD

```

3858	1724	06A0	BL	@OTRCD	OUTPUT RECORD
	1726	1764			
3859				ROOM	
3860	1728	D503	MOV	ABV, *OBJPT	MOVE IN ABSOLUTE MARKER
3861	172A	C320	MOV	@MAINW+RELABS+RELABS, CKPT	IS CODE RELOCATABLE?
	172C	042E			
3862	172E	1302	JEQ	ABOK	NO - SKIP
3863	1730	B520	AB	@EIGHT+1, *OBJPT	MARK AS RELOCATABLE
	1732	0071			
3864			*		
3865			*	MOVE ADDRESS INTO OBJECT CODE	
3866			*		
3867				ABOK	
3868	1734	0584	INC	OBJPT	POINT INTO OBJECT BUFFER
3869	1736	0207	LI	INSPT, LOCNUM	POINT TO ADDRESS IN LINE
	1738	0132			
3870	173A	DD37	MOV	*INSPT+, *OBJPT+	MOVE ADDRESS
3871	173C	DD37	MOV	*INSPT+, *OBJPT+	
3872	173E	DD37	MOV	*INSPT+, *OBJPT+	
3873	1740	DD37	MOV	*INSPT+, *OBJPT+	
3874				OVAL	
3875	1742	8244	C	OBJPT, CDLIM	IS THERE ROOM
3876	1744	14EF	JHE	OUTRCD	NO - OUTPUT RECORD
3877			*		
3878			*	MOVE DATA INTO OBJECT CODE	
3879			*		
3880	1746	D508	MOV	ABDV, *OBJPT	MARK AS ABSOLUTE DATA
3881	1748	9820	CB	@QT, @RELMRK	IS DATA RELOCATABLE?
	174A	1A6B			
	174C	013C			
3882	174E	1602	JNE	ABDAT	NO - SKIP
3883	1750	B520	AB	@Z1, *OBJPT	YES- ADD IN RELOCATABILITY
	1752	006E			
3884				ABDAT	
3885	1754	0584	INC	OBJPT	POINT INTO OBJECT
3886	1756	0207	LI	INSPT, INSVAL	POINT TO DATA
	1758	0138			
3887	175A	DD37	MOV	*INSPT+, *OBJPT+	MOVE IN DATA WORD
3888	175C	DD37	MOV	*INSPT+, *OBJPT+	
3889	175E	DD37	MOV	*INSPT+, *OBJPT+	
3890	1760	DD37	MOV	*INSPT+, *OBJPT+	
3891	1762	0380	RTWP		RETURN
3892			*		
3893			*	OUTPUT RECORD	
3894			*		
3895				OTRCD	
3896	1764	DD02	MOV	SVN, *OBJPT+	COMPUTE CHECKSUM
3897	1766	020C	LI	CKPT, OBJRCD	POINT TO START OF OBJECT
	1768	0078			
3898	176A	04C1	CLR	CKSUM	
3899			*		
3900			*	COMPUTE CHECKSUM	
3901			*		
3902				CKCOM	
3903	176C	04C7	CLR	INSPT	CLEAR CHARACTER

3904	176E	D1FC		MOVB *CKPT+, INSPT	GET CHARACTER
3905	1770	06C7		SWPB INSPT	
3906	1772	A047		A INSPT, CKSUM	ADD IN CHARACTER VALUE
3907	1774	810C		C CKPT, OBJPT	FINISHED?
3908	1776	11FA		JLT CKCOM	
3909	1778	0501		NEG CKSUM	
3910	177A	C804		MOV OBJPT, @CONLOC	CREATE CONVERSION LOCN
	177C	1784'			
3911	177E	0420		BLWP @BINHEX	CONVERT
	1780	0BE0'			
3912	1782	16E8'		DATA CKSM	CHECKSUM VALUE
3913	1784	0000	CONLOC	DATA 0	INTO OBJECT BUFFER
3914	1786	0224		AI OBJPT, 4	ADJUST OBJECT POINTER
	1788	0004			
3915	178A	DD06		MOVB FVL, *OBJPT+	PUT OUT 'F'
3916	178C	D520		MOVB @CR, *OBJPT	TERMINATE WITH CR
	178E	0101'			
3917	1790	0204		LI OBJPT, OBJRCD	INITIALIZE OBJECT POINTER
	1792	0078'			
3918			OTWRT		
3919	1794	C284		MOV OBJPT, IOPARM	
3920	1796	0420		BLWP @WRITE	WRITE RECORD TO OBJECT FILE
	1798	19C2'			
3921	179A	1309		JEQ CLNBUF	CONTINUE, IF NO ERROR
3922	179C	020A		LI IOPARM, RDYMSG	PRINT MESSAGE TO READY OBJ
	179E	0119'			
3923	17A0	0420		BLWP @PRINTN	
	17A2	19D6'			
3924	17A4	020A		LI IOPARM, INSBUF	WAIT FOR RESPONSE
	17A6	017C'			
3925	17A8	0420		BLWP @KEYIN	
	17AA	198A'			
3926	17AC	10F3		JMP OTWRT	TRY AGAIN
3927			*		
3928			*	CLEAR OBJECT BUFFER	
3929			*		
3930			CLNBUF		
3931	17AE	CD20		MOV @BLANK, *OBJPT+	BLANK OUT RECORD
	17B0	17DC'			
3932	17B2	0284		CI OBJPT, OBJRCD+71	
	17B4	00BF'			
3933	17B6	11FB		JLT CLNBUF	
3934	17B8	0204		LI OBJPT, OBJRCD	INITIALIZE OBJECT POINTER
	17BA	0078'			
3935	17BC	045B		RT	
3936			CLROBJ		
3937	17BE	16E6'		DATA OTWP, CLRBG	WORKSPACE, START
	17C0	17C2'			
3938			CLRBG		
3939	17C2	06A0		BL @OTRCD	OUTPUT OBJECT RECORD
	17C4	1764'			
3940	17C6	0380		RTWP	RETURN

3943	17C8	0000	OTBWP	DATA 0	WR0	INSPT2
3944	17CA	4100		DATA >4100	WR1	ABET
3945	17CC	0100		DATA >100	WR2	AODD
3946	17CE	013A		DATA INSVAL+2	WR3	SBYT
3947	17D0	0138		DATA INSVAL	WR4	FBYT
3948	17D2	3030		DATA >3030	WR5	ZZERO
3949	17D4	0000		DATA 0	WR6	SHLD
3950	17D6	0000		DATA 0	WR7	OBJPT2
3951	17D8	0000		DATA 0	WR8	BYTFL
3952	17DA	1706		DATA OUTOBJ	WR9	OTOBPT
3953	17DC	2020	BLANK	DATA >2020	WR10	BLNNK
3954	17DE	0134		DATA LOCNUM+2	WR11	LOCP2
3955	17E0	0135		DATA LOCNUM+3	WR12	LOCP3
3956	17E2	0000		DATA 0	WR13	OLD WP
3957	17E4	0000		DATA 0	WR14	OLD PC
3958	17E6	0000		DATA 0	WR15	OLD SR


```

3960 * TITLE: OUTBYT
3961 * OUTPUT OBJECT BYTE
3962 * REVISION: 03/01/74
3963 * ORIGINAL RELEASE
3964 * 03/15/76
3965 * MODIFIED TO RUN WITH PXRMT, DORG PROCESSING
3966 * ADDED
3967 * COMPUTER: 990, ASSEMBLY
3968 * ABSTRACT:
3969 * THIS ROUTINE COORDINATES THE PLACEMENT OF BYTE
3970 * DATA INTO THE OBJECT CODE. IF THE BYTE OCCURS
3971 * ON A WORD BOUNDARY, IT IS PLACED INTO THE OBJECT
3972 * AS A LEFT-JUSTIFIED, ZERO-FILLED WORD. IF THE
3973 * BYTE OCCURS ON A NON-WORD BOUNDARY IN
3974 * CONSECUTIVE CODE, THE BYTE IS 'OR'ED INTO THE
3975 * OBJECT WORD PRESENTLY THERE. IF IN NON-
3976 * CONSECUTIVE CODE, THE BYTE IS PLACED IN THE
3977 * OBJECT AS A RIGHT-JUSTIFIED, ZERO-FILLED WORD.
3978 * CALLING SEQUENCE:
3979 * BLWP @OUTBYT
3980 * ADDITIONAL NOTES:
3981 * WORKSPACE = OTBWP (UNSHARED)
3982 * ROUTINES CALLED: OUTOBJ
3983 OUTBYT
3984 17E8 17C8' DATA OTBWP WORKSPACE
3985 17EA 17EC' DATA OTBBG START
3986 OTBBG
3987 17EC C1A0 MOV @DORGFL,SHLD EXIT IF IN DORG
3988 17EE 11C8'
3989 17F0 1301 JEQ OUTB
3990 17F2 0380 RTWP
3991 OUTB
3992 *
3993 * CHECK FOR CONSECUTIVE CODE AND BOUNDARY LOCATION
3994 *
3994 17F4 C1A0 MOV @OTWP+NCONS+NCONS,SHLD CONSECUTIVE CODE?
3995 17F6 16E6'
3995 17F8 1318 JEQ CONPT YES
3996 17FA D01C MOVB *LOCP3,INSPT2 NO, GET LAST BYTE OF ADDRESS
3997 17FC 9040 CB INSPT2,ABET DECIDE IF BYTE EVEN/ODD
3998 17FE 1103 JLT DECM DECIMAL VALUE (0-9)
3999 1800 2002 COC ADD,INSPT2
4000 1802 160A JNE ODD BYTE NOT ON WORD BOUNDARY
4001 1804 1002 JMP EVN BYTE ON WD BOUNDARY
4002 DECM
4003 1806 2002 COC ADD,INSPT2
4004 1808 1307 JEQ ODD BYTE ODD
4005 *
4006 * OUTPUT WORD BOUNDARY BYTE
4007 *
4008 EVN
4009 180A C513 MOV *SBYT,*FBYT OUTPUT EVEN BYTE
4010 180C C4C5 MOV ZZERO,*SBYT FAKE WORD OBJECT
4011 180E 0419 BLWP *OTOBPT OUTPUT OBJECT
  
```

4012	1810	C4D4	MOV	*Fbyt,*sbyt	RESTORE
4013	1812	C50A	MOV	BLNnk,*fbyt	BLANK OUT FIRST BYTE
4014	1814	0588	INC	BYTFL	SET BYTE ADDRESS FLAG
4015	1816	0380	RTWP		RETURN
4016			*		
4017			*	OUTPUT NON-WORD BOUNDARY BYTE	
4018			*		
4019			ODD		
4020	1818	C193	MOV	*sbyt,shld	SAVE TRUE VALUE
4021	181A	C505	MOV	ZZERO,*fbyt	OUTPUT ZERO WORD
4022	181C	C4C5	MOV	ZZERO,*sbyt	
4023	181E	061B	DEC	*LOCP2	SET EVEN ADDRESS
4024	1820	0419	BLWP	*OTOBPT	OUTPUT OBJECT
4025	1822	059B	INC	*LOCP2	RESTORE ADDRESS
4026	1824	C50A	MOV	BLNnk,*fbyt	RESTORE VALUE
4027	1826	C4C6	MOV	shld,*sbyt	PLACE VALUE IN BYTE
4028	1828	1008	JMP	CNOD	OUTPUT ODD BYTE
4029			*		
4030			*	CONSECUTIVE CODE PROCESSING	
4031			*		
4032			CONPT		
4033	182A	D01C	MOVb	*LOCP3,INSPT2	TEST FOR WD BOUNDARY
4034	182C	9040	CB	INSPT2,ABET	
4035	182E	1103	JLT	DECM2	DECIMAL VALUE (0-9)
4036	1830	2002	COC	ADD,INSPT2	
4037	1832	13EB	JEQ	EVN	WD BOUNDARY
4038	1834	1002	JMP	CNOD	ODD BOUNDARY
4039			DECM2		
4040	1836	2002	COC	ADD,INSPT2	
4041	1838	16E8	JNE	EVN	WD BOUNDARY
4042			CNOD		
4043	183A	C1E0	MOV	@OTWP+OBJPT+OBJPT,OBJPT2	POINT INTO OBJECT
	183C	16EE			
4044	183E	0647	DECT	OBJPT2	POINT TO STUFFING ADDRESS
4045	1840	D0F3	MOVb	*sbyt+,*objpt2+	PLACE BYTE INTO WORD
4046	1842	D5D3	MOVb	*sbyt,*objpt2	
4047	1844	0603	DEC	sbyt	RESTORE POINTER TO BYTE
4048	1846	04C8	CLR	BYTFL	CLEAR BYTE BOUNDARY FLAG
4049	1848	0380	RTWP		RETURN

```
4052      *
4053      * I/O INTERFACE WORKSPACE AREA
4054      *
4055      LNREG
4056  184A  0BC0'  DATA BINVL      R0  BINVLP
4057  184C  012A'  DATA IOBUF     R1  PRBF
4058  184E  017C'  DATA INSBUF    R2  RDBF
4059  1850  0000    DATA 0        R3  PRTPOS
4060  1852  17DC'  DATA BLANK     R4  BLPT
4061  1854  013C'  DATA RELMRK    R5  RMK
4062  1856  0000  LIST  DATA 0      R6  LSTPT
4063  1858  0000  FSTPT DATA 0      R7  FST
4064  185A  0000    DATA 0        R8  TEMPRG
4065  185C  0000    DATA 0        R9
4066  185E  0000    DATA 0        R10 I/O RTN STATUS
4067  1860  15AE'  DATA PGHDR     R11 PGHC
4068  1862  0000  RCDCNT DATA 0     R12 RCD1
4069  1864  0000    DATA 0        R13 OLD WP
4070  1866  0000    DATA 0        R14 OLD PC
4071  1868  0000    DATA 0        R15 OLD SR
```

```

4073 *
4074 * TITLE: PRNTLN/RDRCD
4075 * I/O INTERFACE
4076 * REVISION: 03/01/74
4077 * ORIGINAL RELEASE
4078 * 03/15/76
4079 * MODIFIED TO RUN WITH PXRMTN
4080 * COMPUTER: 990, ASSEMBLY
4081 * ABSTRACT:
4082 * TWO ROUTINES (PRNTLN,RDRCD) ARE USED TO PRINT
4083 * SOURCE LINES AND READ IN SOURCE RECORDS. PRNTLN
4084 * ONLY PRINTS THE SOURCE LINE ONCE, EVEN IF IT
4085 * RESULTS IN SEVERAL LINES OF OBJECT OUTPUT
4086 * RDRCD BLANKS THE INPUT AND OUTPUT LINES PRIOR
4087 * TO EACH SOURCE RECORD READ. PRNTLN COUNTS THE
4088 * LINES.
4089 * CALLING SEQUENCE:
4090 * BLWP @PRNTLN OR BLWP @RDRCD
4091 * ADDITIONAL NOTES:
4092 * WORKSPACE = LNREG (UNSHARED)
4093 * ROUTINES CALLED: PGHDR, BINDEC, PRINTP, READ,
4094 * ENDFIL
4095 PRNTLN
4096 186A 184A' DATA LNREG,PRT WORKSPACE,START
      186C 186E'
4097 PRT
4098 186E C186 MOV LSTPT,LSTPT IS LIST FLAG SET?
4099 1870 133D JEQ OVR NO, SKIP
4100 1872 0620 DEC @LNCNT YES, DECR LINE COUNT
      1874 1590'
4101 1876 1501 JGT BGPRT LESS THAN 55 LINES, CONTINUE
4102 1878 041B BLWP *PGHC MORE THAN 55, PRINT PAGE HDR
4103 BGPRT
4104 187A C1C7 MOV FST,FST FIRST TIME FOR LINE?
4105 187C 133B JEQ INPT NO, SKIP
4106 187E C40C MOV RCD1,*BINVLP YES, CONVERT RECORD NUMBER
4107 1880 0420 BLWP @BINDEC TO DECIMAL AND PLACE
      1882 0BF4'
4108 1884 012C' DATA RCDNUM IN INSTRUCTION BUFFER
4109 *
4110 * TRANSFER CHARS FROM READ BUFF TO PRINT BUFF
4111 * EXPANDING TABS AND INSERTING LF
4112 *
4113 1886 04C3 CLR PRTPOS INIT PRINT POSITION
4114 1888 04CA CLR R10 INIT BUFF PTR
4115 188A 04C8 CLR TEMPRG INIT LAST NON-BLANK ADDR
4116 PRBLD
4117 188C D272 MOVB *RDBF+,R9 GET CHAR FROM READ BUFFER
4118 188E 9809 CB R9,@TAB IS IT A TAB?
      1890 0004'
4119 1892 130E JEQ PRTAB YES
4120 1894 9809 CB R9,@CR IS IT A CR?
      1896 0101'
4121 1898 131D JEQ PROCR YES
  
```

```

4122 189A DA89          MOVB R9,@PRBUF(R10)    NO, STORE CHAR IN PRINT BUFF
      189C 013E'
4123 189E 058A          INC R10                 INCR BUFF PTR
4124 18A0 9509          CB R9,*BLPT            IS IT A BLANK?
4125 18A2 1304          JEQ PRBL2             YES, SKIP
4126 18A4 C20A          MOV R10,TEMPRG        SAVE LAST NON-BLANK ADDR
4127 18A6 028A          CI R10,59            MORE THAN 59 CHARS?
      18A8 003B
4128 18AA 1514          JGT PRCR              YES, THEN EXIT
4129                                PRBL2
4130 18AC 0583          INC PRTPOS            INCR PRINT POSITION
4131 18AE 10EE          JMP PRBLD            CONTINUE
4132                                *
4133                                * EXPAND TABS
4134                                *
4135                                PRTAB
4136 18B0 04C9          CLR R9                INSERT SPACES TO NEXT TAB
4137                                PRTB1
4138 18B2 8A43          C PRTPOS,@TABS(R9)   FIND TAB GREATER THAN
      18B4 1598'
4139 18B6 1104          JLT PRTSP             CURRENT PRINT POSITION
4140 18B8 05C9          INCT R9              INCR TAB BUFF INDEX
4141 18BA 0289          CI R9,8              UP TO 4 TABS ALLOWED
      18BC 0008
4142 18BE 16F9          JNE PRTB1
4143                                PRTSP
4144 18C0 DA94          MOVB *BLPT,@PRBUF(R10) FOUND, INSERT SPACES
      18C2 013E'
4145 18C4 058A          INC R10              UP TO NEXT TAB STOP
4146 18C6 0583          INC PRTPOS           INCR PRINT POSITION
4147 18C8 8A43          C PRTPOS,@TABS(R9)  PRINT POSITION = TAB STOP
      18CA 1598'
4148 18CC 11F9          JLT PRTSP             ELSE INSERT ANOTHER SPACE
4149 18CE 028A          CI R10,60           LESS THAN 60 CHARS?
      18D0 003C
4150 18D2 11DC          JLT PRBLD            YES, THEN CONTINUE
4151                                *
4152                                PRCR
4153 18D4 DA20          MOVB @LF,@PRBUF(TEMPRG) TERM WITH LF/CR
      18D6 0100'
      18D8 013E'
4154 18DA DA20          MOVB @CR,@PRBUF+1(TEMPRG)
      18DC 0101'
      18DE 013F'
4155                                OGO
4156 18E0 0202          LI R0BF,INSBUF       REINIT READ BUFFER ADDR
      18E2 017C'
4157 18E4 04C7          CLR FST              CLEAR FIRST TIME FLAG
4158 18E6 C281          MOV PRBF,IOPARM      POINT TO PRINT BUFFER
4159 18E8 0420          BLWP @PRINTP         PRINT LINE
      18EA 19D2'
4160                                OVR
4161 18EC D554          MOVB *BLPT,*RMK      BLANK RELOCATABILITY MARK
4162 18EE D814          MOVB *BLPT,@LOCNUM+4 BLANK POSSIBLE '***'
      18F0 0136'

```

```

4163 18F2 0380          RTWP          RETURN
4164                    *
4165                    * SECOND AND SUBSEQUENT CALLS FOR PRINT
4166                    *
4167                    INPT
4168 18F4 D820          MOVB @LF,@RELMRK+1    PLACE LF IN LINE
      18F6 0100
      18F8 013D
4169 18FA D820          MOVB @CR,@RELMRK+2    PLACE CR IN LINE
      18FC 0101
      18FE 013E
4170 1900 C814          MOV  *BLPT,@RCDNUM    BLANK OUT RECORD NUMBER
      1902 012C
4171 1904 C814          MOV  *BLPT,@RCDNUM+2
      1906 012E
4172 1908 10EB          JMP  OGO
4173                    *
4174                    * READ RECORD
4175                    *
4176                    RDRCD
4177 190A 184A          DATA LNREG, RD      WORKSPACE. RD
      190C 190E
4178                    RD
4179 190E C282          MOV  RDBF, IOPARM      POINT TO READ BUFFER
4180 1910 C201          MOV  PRBF, TEMPRG     POINT TO TOTAL BUFFER
4181                    RD2
4182 1912 CE14          MOV  *BLPT,*TEMPRG+   CLEAR OUT BUFFER
4183 1914 0288          CI   TEMPRG, IOBUF+144
      1916 01BA
4184 1918 11FC          JLT  RD2
4185                    *
4186                    * READ RECORD
4187                    *
4188 191A 0420          BLWP @READ            READ IN RECORD
      191C 1980
4189 191E 0249          ANDI R9, EOF+IOERR   EOF OR I/O ERROR ENCOUNTERED
      1920 6000
4190 1922 1605          JNE  WREOF            YES, GO WRITE EOF
4191 1924 DE94          MOVB *BLPT,*IOPARM+  TERMINATE RECORD WITH
4192 1926 D6A0          MOVB @CR,*IOPARM     BLANK/CR
      1928 0101
4193 192A 058C          INC  RCD1             INCREMENT RECORD COUNT
4194 192C 0380          RTWP                RETURN
4195                    *
4196                    * WRITE END OF FILE
4197                    *
4198                    WREOF
4199 192E 02E0          LWPI IDWKS           USE I/O ROUTINE WORKSPACE
      1930 1944
4200 1932 04C2          CLR  IOC              INIT CODE FOR I/O SUPV CALL
4201 1934 C0E0          MOV  @C2LUNG, IOPLUN STORE LUNG
      1936 0062
4202 1938 D0E0          MOVB @EOFCOD, IOPLUN STORE I/O OP
      193A 0067
4203 193C 04C4          CLR  FLGS            INIT FLAGS
  
```

PXRASM
I/O INTERFACE

MIRA990 V2L1 17:44:10
948925-9901**

197/76

PAGE 0115

4204 193E 2FC2
4205 1940 0460
1942 0450

SVC IOC
B @PXRASM

MAKE SUPV CALL
YES, RESTART

4208
4209
4210
4211
4212 1944 0900
4213 1946 0B00
4214 1948
4215 1958 003C
4216 195A

*
* WORKSPACE AREA FOR INPUT/OUTPUT ROUTINES
*

IOWKS

DATA RDCOD	R0 RDCODP
DATA WRCOD	R1 WRCODP
SCBLK BSS 16	R2-R9 I/O SUPV CALL BLK (PRB)
DATA 60	R10 LEN
BSS 10	R11-R15


```

4218          * TITLE:      OPEN
4219          *              ASSIGN A DEVICE TO A TASK
4220          * REVISION    03/15/76
4221          *              ORIGINAL
4222          * COMPUTER:   990, ASSEMBLY
4223          * ABSTRACT:
4224          *              SETS UP PRB AND MAKES SUPERVISOR CALL TO OPEN
4225          *              A DEVICE
4226          * CALLING SEQUENCE:
4227          *              R10=LUN0
4228          *              BLWP @OPEN
4229          * ADDITIONAL NOTES:
4230          *              WORKSPACE = IOWKS (SHARED WITH I/O ROUTINES)
4231          OPEN
4232          1964 1944'      DATA IOWKS, $+2      TRANSFER VECTOR
              1966 1968'
4233          1968 04C3      CLR IOPLUN
4234          196A D0E0      MOVB @OPNCOD, IOPLUN      STORE I/O OP
              196C 0066'
4235          196E 04C4      CLR FLGS              INIT FLAGS
4236          1970 04C2      CLR IOC              INIT CODE FOR I/O SUPV CALL
4237          1972 C30D      MOV R13, R12         GET CALLER'S WORKSPACE PTR
4238          1974 022C      AI R12, R10+R10      INDEX TO CALL PARAMETER
              1976 0014
4239          1978 C31C      MOV *R12, R12        GET LUN0
4240          197A E0CC      SOC R12, IOPLUN       OR LUN0 INTO IOPLUN
4241          197C 2FC2      SVC IOC              MAKE SUPV CALL
4242          197E 0380      RTWP                 RETURN

```

```

4244 * TITLE: READ
4245 * READ A RECORD
4246 * REVISION: 03/15/76
4247 * ORIGINAL
4248 * COMPUTER: 990, ASSEMBLY
4249 * ABSTRACT:
4250 * SETS UP PRB AND MAKES SUPERVISOR CALL TO READ
4251 * A RECORD
4252 * CALLING SEQUENCE:
4253 * R10=BUFFER ADDRESS
4254 * BLWP @READ
4255 * R10=BUFFER ADDRESS+CHAR COUNT
4256 * R9=FLAGS
4257 * ADDITIONAL NOTES:
4258 * WORKSPACE = IOWKS (SHARED WITH I/O ROUTINES)
4259 * RETURN PARAMETERS:
4260 * R10 = BUFFER ADDR+CHAR COUNT
4261 * R9 = FLAGS
4262 READ
4263 1980 1944' DATA IOWKS, $+2 TRANSFER VECTOR
4264 1982 1984'
4264 1984 C0E0 MOV @C1LUN0, IOPLUN STORE LUN0
4265 1986 0060'
4265 1988 1004 JMP KEYRD
4266 *
4267 * TITLE: KEYIN
4268 * KEYBOARD INPUT ROUTINE
4269 * REVISION: 03/15/76
4270 * ORIGINAL
4271 * COMPUTER: 990, ASSEMBLY
4272 * ABSTRACT:
4273 * SETS UP PRB AND MAKES SUUPERVISOR CALL TO INPUT
4274 * A RECORD FROM KEYBOARD
4275 * CALLING SEQUENCE:
4276 * R10=BUFFER ADDRESS
4277 * BLWP @KEYIN
4278 * ADDITIONAL NOTES:
4279 * WORKSPACE = IOWKS (SHARED WITH I/O ROUTINES)
4280 * RETURN PARAMETERS:
4281 * R10 = BUFFER ADDRESS+CHAR COUNT
4282 * R9 = FLAGS
4283 KEYIN
4284 198A 1944' DATA IOWKS, $+2 TRANSFER VECTOR
4285 198C 198E'
4285 198E C0E0 MOV @KBLUN0, IOPLUN STORE LUN0
4286 1990 005C'
4286 KEYRD
4287 1992 D0C0 MOVB RDCODP, IOPLUN STORE I/O OP
4288 1994 C300 MOV R13, R12 GET CALLER'S WORKSPACE PTR
4289 1996 022C AI R12, R10+R10 INDEX TO CALLING PARAMETERS
4290 1998 0014
4290 199A C150 MOV *R12, BUFADR SET UP BUFFER ADDRESS
4291 199C C18A MOV LEN, BUFLen SET UP BUFFER LENGTH
4292 199E 04C4 CLR FLGS INIT FLAGS

```

4293	19A0	04C7	CLR	CHRCNT	INIT CHAR CNT TO ZERO
4294	19A2	04C2	CLR	IOC	INIT CODE FOR I/O SUPV CODE
4295	19A4	2FC2	SVC	IOC	MAKE SUPV CALL
4296	19A6	A707	A	CHRCNT, *R12	INCR BUFF PTR TO END OF DATA
4297	19A8	0243	ANDI	IOPLUN, >00FF	IS THIS KEYBOARD READ?
	19AA	00FF			
4298	19AC	8803	C	IOPLUN, @KBLUN0	
	19AE	005C			
4299	19B0	1303	JEQ	KEYPLF	IF YES, GO PRINT LF
4300	19B2	064C	DECT	R12	MOVE FLAGS TO CALLING R9
4301	19B4	C704	MOV	FLGS, *R12	TO RETURN READ STATUS
4302	19B6	0380	RTWP		ELSE, RETURN
4303					
4304	19B8	D0C1	MOVB	WRCODP, IOPLUN	SET UP PRB TO PRINT LF
4305	19BA	0205	LI	BUFADR, LFCR	SET UP BUFFER ADDR
	19BC	0100			
4306	19BE	0704	SETO	FLGS	SET FLAG TO PRINT WITHOUT CR
4307	19C0	101A	JMP	PRTKEY	JUMP TO ENTRY POINT IN PRINT

```

4309      * TITLE:      WRITE
4310      *              WRITE A RECORD
4311      * REVISION:   03/15/76
4312      *              ORIGINAL
4313      * COMPUTER:   990, ASSEMBLY
4314      * ABSTRACT:
4315      *              SETS UP PRB AND MAKES SUPERVISOR CALL TO WRITE
4316      *              A RECORD
4317      * CALLING SEQUENCE:
4318      *              R10=BUFFER ADDRESS
4319      *              BLWP @WRITE
4320      * ADDITIONAL NOTES:
4321      *              WORKSPACE = IOWKS (SHARED WITH I/O ROUTINES)
4322      WRITE
4323      19C2 1944'      DATA IOWKS, $+2      TRANSFER VECTOR
4324      19C4 19C6'
4324      19C6 C0E0      MOV @C2LUN0, IOPLUN      STORE LUN0
4324      19C8 00E2'
4325      19CA 0704      SET0 FLGS      SET FLAG FOR WRITE WITHOUT CR
4326      19CC 100F      JMP PRWRT
4327
4328      * TITLE:      PRINT/PRINTN/PRINTP
4329      *              PRINT A RECORD
4330      * REVISION:   03/15/76
4331      *              ORIGINAL
4332      * COMPUTER:   990, ASSEMBLY
4333      * ABSTRACT:   SETS UP PRB AND MAKES SUPV CALL
4334      *              TO PRINT A RECORD
4335      *              THREE ENTRY POINTS:
4336      *              PRINT - PRINTS RECORD WITH CR TO LUN0 0
4337      *              PRINTN - PRINTS RECORD WITHOUT CR TO LUN0 0
4338      *              PRINTP - PRINTS RECORD WITH CR TO LUN0 6
4339      * CALLING SEQUENCE:
4340      *              R10 = BUFFER ADDRESS
4341      *              BLWP @PRINT
4342      *              BLWP @PRINTN
4343      *              BLWP @PRINTP
4344      * STATISTICS: WORKSPACE = IOWKS (SHARED WITH I/O ROUTINES)
4345      *
4346      19CE 1944'      PRINT DATA IOWKS, PRTCR
4346      19D0 19DA'
4347      19D2 1944'      PRINTP DATA IOWKS, PRTPCR
4347      19D4 19DE'
4348      19D6 1944'      PRINTN DATA IOWKS, PRTNCR
4348      19D8 19E6'
4349      PRTCR
4350      19DA 04C4      CLR FLGS      SET FLAG FOR PRINT WITH CR
4351      19DC 1005      JMP PRENT
4352      PRTPCR
4353      19DE 04C4      CLR FLGS      SET FLAG FOR PRINT WITH CR
4354      19E0 C0E0      MOV @PRLUN0, IOPLUN      SET LUN0 6 FOR PRINT
4354      19E2 005E'
4355      19E4 1003      JMP PRWRT
4356      PRTNCR

```

4357	19E6	0704	SET0	FLGS	SET FLAG FOR PRINT WITHOUT CR
4358			PRENT		
4359	19E8	C0E0	MOV	@KBLUN0, IOPLUN	SET LUN0 0 FOR PRINT
	19EA	005C			
4360			PRWRT		
4361	19EC	D0C1	MOVB	WRCODP, IOPLUN	STORE I/O OP
4362	19EE	C30D	MOV	R13, R12	GET CALLER'S WP
4363	19F0	022C	AI	R12, R10+R10	INDEX TO CALLING PARAMETERS
	19F2	0014			
4364	19F4	C15C	MOV	*R12, BUFADR	SET UP BUFFER ADDR
4365			PRTKEY		
4366	19F6	06A0	BL	@GETCNT	GET CHAR COUNT
	19F8	1A1A			
4367	19FA	A1C4	A	FLGS, CHRCNT	ADJUST CHRCNT FOR CR
4368	19FC	04C4	CLR	FLGS	INIT FLAGS
4369	19FE	04C2	CLR	IOC	INIT CODE FOR I/O SUPV CALL
4370	1A00	2FC2	SVC	IOC	MAKE SUPV CALL
4371	1A02	0244	ANDI	FLGS, IOERR	CHECK RETURN STATUS
	1A04	4000			
4372	1A06	02CF	STST	R15	STORE FLAGS IN USER STATUS RE
4373	1A08	1307	JEQ	PREXT	ZERO, EXIT
4374	1A0A	0243	ANDI	IOPLUN, >00FF	
	1A0C	00FF			
4375	1A0E	8803	C	IOPLUN, @PRLUN0	PRINTER ERR (ESC CNTL RTN)
	1A10	005E			
4376	1A12	1602	JNE	PREXT	NO
4377	1A14	0460	B	@PXRASM	YES, RESTART ASSEMBLY
	1A16	0450			
4378			PREXT		
4379	1A18	0300	RTWP		RETURN

```
4381 * TITLE: GETCNT
4382 * GET CHARACTER COUNT
4383 * REVISION: 03/15/76
4384 * ORIGINAL
4385 * COMPUTER: 990, ASSEMBLY
4386 * ABSTRACT:
4387 * SCANS LINE COUNTING CHARACTERS UP TO AND
4388 * INCLUDING CR. RETURNS COUNT IN CHRCNT (R7)
4389 * CALLING SEQUENCE:
4390 * R10=BUFFER ADDRESS
4391 * BL @GETCNT
4392 * ADDITIONAL NOTES:
4393 * RETURN PARAMETERS:
4394 * CHRCNT (R7) = CHARACTER COUNT
4395 GETCNT
4396 1A1A 04C7 CLR CHRCNT INIT CHAR COUNT
4397 GCLOOP
4398 1A1C 0587 INC CHRCNT INCR CHAR COUNT
4399 1A1E 9835 CB *BUFADR+,@CR COMPARE CHAR TO CR
4400 1A20 0101
4401 1A22 1303 JEQ GCEXT IF EQUAL, EXIT
4401 1A24 0287 CI CHRCNT,82 MORE THAN 80 CHARS + LF/CR?
4402 1A26 0052
4402 1A28 11F9 JLT GCLOOP NO. CONTINUE
4403 GCEXT
4404 1A2A 6147 S CHRCNT,BUFADR RESET BUFFER ADDR TO START
4405 1A2C 045B B *RTN RETURN
```

```

4407      *
4408      *      JUMP TABLES USING CHARACTER CLASS
4409      *
4410      JMPTB1
4411  1A2E    00      BYTE ALPHA-JMP1/2-1      ALPHA
4412  1A2F    22      BYTE NUM-JMP1/2-1        NUM
4413  1A30    6F      BYTE DOLLAR-JMP1/2-1      $
4414  1A31    5C      BYTE QUOTE-JMP1/2-1      QUOTE
4415  1A32    34      BYTE HEX-JMP1/2-1          >
4416
4417      JMPTB2
4417  1A33    00      BYTE AL-JMP2/2-1          AL
4418  1A34    05      BYTE NU-JMP2/2-1          NU
4419  1A35    EC      BYTE NUMO-JMP2/2-1       NUMO
4420  1A36    EC      BYTE NUMO-JMP2/2-1       NUMO
4421  1A37    EC      BYTE NUMO-JMP2/2-1       NUMO
4422
4423      JMPTB3
4423  1A38    00      BYTE CKER-JMP3/2-1      0
4424  1A39    18      BYTE OPSTK1-JMP3/2-1    1
4425  1A3A    1B      BYTE OPSTK2-JMP3/2-1    2
4426  1A3B    1F      BYTE OPSTK3-JMP3/2-1    3
4427  1A3C    1F      BYTE OPSTK3-JMP3/2-1    4
4428  1A3D    3B      BYTE SCAL-JMP3/2-1      5
4429  1A3E    3B      BYTE SCAL-JMP3/2-1      6
4430  1A3F    3B      BYTE SCAL-JMP3/2-1      8
4431  1A40    27      BYTE SCQT-JMP3/2-1      7
4432  1A41    3B      BYTE SCAL-JMP3/2-1      9
4433
4434      *
4435      *      JUMP TABLES USING SYMBOL TYPE
4436      *
4437      JMPTB6
4437  1A42    00      BYTE RDF-JMP6/2-1      0
4438  1A43    00      BYTE RDF-JMP6/2-1      1
4439  1A44    01      BYTE ADF-JMP6/2-1      2
4440  1A45    01      BYTE ADF-JMP6/2-1      3
4441  1A46    00      BYTE RDF-JMP6/2-1      4
4442  1A47    00      BYTE RDF-JMP6/2-1      5
4443  1A48    01      BYTE ADF-JMP6/2-1      6
4444  1A49    01      BYTE ADF-JMP6/2-1      7
4445  1A4A    18      BYTE ERRS-JMP6/2-1     8
4446  1A4B    06      BYTE ED-JMP6/2-1      9
4447  1A4C    06      BYTE ED-JMP6/2-1     10
4448  1A4D    0E      BYTE FR-JMP6/2-1     11
4449  1A4E    24      BYTE ER-JMP6/2-1     12
4450  1A4F    24      BYTE ER-JMP6/2-1     13
4451
4452      FTAB
4452  1A50    00      BYTE EXTDFR-JMPF/2-1  FLAGS = 0
4453  1A51    2E      BYTE NXSYM-JMPF/2-1  FLAGS = 1
4454  1A52    03      BYTE EXTDFR-JMPF/2-1  FLAGS = 2
4455  1A53    2B      BYTE NXSYM-JMPF/2-1  FLAGS = 3
4456  1A54    00      BYTE EXTDFR-JMPF/2-1  FLAGS = 4
4457  1A55    2E      BYTE NXSYM-JMPF/2-1  FLAGS = 5
4458  1A56    03      BYTE EXTDFR-JMPF/2-1  FLAGS = 6
4459  1A57    2B      BYTE NXSYM-JMPF/2-1  FLAGS = 7
4460  1A58    2E      BYTE NXSYM-JMPF/2-1  FLAGS = 8

```

I/O ROUTINES

948925-9901**

```

4461 1A59 3E          BYTE UNDSM-JMPF/2-1    FLAGS = 9
4462 1A5A 3E          BYTE UNDSM-JMPF/2-1    FLAGS = 10
4463 1A5B 3E          BYTE UNDSM-JMPF/2-1    FLAGS = 11
4464 1A5C 3B          BYTE EXTRFA-JMPF/2-1   FLAGS = 12
4465 1A5D 3B          BYTE EXTRFR-JMPF/2-1   FLAGS = 13
4466
4467 *
4468 *          JUMP TABLE USING INSTRUCTION
4469 *          WORD LENGTH AND RELOCATABILITY
4470
4471 1A5E 02          INTBL
4472 1A5F 01          BYTE IN0-JMPIN/2-1    NO EXTRA WDS
4473 1A60 04          BYTE IN1-JMPIN/2-1    1 EX WD -DEF'D ABS      1
4474 1A61 07          BYTE IN2-JMPIN/2-1    1 EX WD -DEF'D REL      2
4475 1A62 00          BYTE IN3-JMPIN/2-1    1 EX WD -FRF             3
4476 1A63 03          BYTE IN4-JMPIN/2-1    (DEF ABS, DEF ABS)      4
4477 1A64 06          BYTE IN5-JMPIN/2-1    (DEF ABS, DEF REL)      5
4478 1A65 09          BYTE IN6-JMPIN/2-1    (DEF ABS, FRF)          6
4479 1A66 0B          BYTE IN7-JMPIN/2-1    (DEF REL, DEF ABS)      7
4480 1A67 0D          BYTE IN8-JMPIN/2-1    (DEF REL, DEF REL)      8
4481 1A68 0F          BYTE IN9-JMPIN/2-1    (DEF REL, FRF)          9
4482 1A69 11          BYTE IN10-JMPIN/2-1   (FRF , DEF ABS)        10
4483 1A6A 13          BYTE IN11-JMPIN/2-1   (FRF , DEF REL)        11
4484 1A6B 27          BYTE IN12-JMPIN/2-1   (FRF, FRF)             12
4485 *          QT

```



```

4488      * TITLE:      FRT/SYMT/INIT
4489      *              FORWARD REFERENCE TABLE/SYMBOL TABLE/HEADING
4490      * REVISION:   03/01/74
4491      *              ORIGINAL
4492      *              03/15/76
4493      *              HEADING AND MEM SIZE INIT MOVED & MODIFIED
4494      *              TO RUN WITH PXRMR, PREDEFINED REG OPTION ADDED
4495      *              12/07/76
4496      *              MEM SIZE REMOVED.
4497      * COMPUTER:   990, ASSEMBLY
4498      * ABSTRACT:   THE AREA FROM END OF PXRASM TO END OF
4499      *              MEMORY RESERVED FOR FORWARD REFERENCE
4500      *              TABLE AND SYMBOL TABLE. FRT STARTS AT
4501      *              BOTTOM OF BUFFER AND IS BUILT UP, SYMT
4502      *              STARTS AT TOP OF SAME BUFFER AND IS
4503      *              BUILT DOWN.
4504      *              INIT CONTAINS CODE FOR INITIALIZATION
4505      *              DONE FIRST TIME PXRASM EXECUTED ONLY.
4506      *
4507      *              FORWARD REFERENCE TABLE
4508      *              AND SYMBOL TABLE
4509      *
4510      00A0 FRT      EQU  >A0
4511      *              REF  SYMT
4512      *
4513      *              RORG FRT
4514      INIT
4515      *
4516      *              INITIALIZE XOP VECTOR
4517      REF  SVCWP
4518      REF  SVCSR
4519      1A6C 0209      LI   R9, SVCWP
4520      1A6E 0000
4521      1A70 C809      MOV  R9, @>7C
4522      1A72 007C
4523      1A74 0209      LI   R9, SVCSR
4524      1A76 0000
4525      1A78 C809      MOV  R9, @>7E
4526      1A7A 007E
4527      *
4528      *              PRINT OUT HEADING
4529      1A7C C2A0      MOV  @KBLUNO, IOPARM      OPEN LOG KEYBOARD
4530      1A7E 005C
4531      1A80 0420      BLWP @OPEN
4532      1A82 1964
4533      1A84 C2A0      MOV  @PRLUNO, IOPARM      OPEN LOG PRINTER
4534      1A86 005E
4535      1A88 0420      BLWP @OPEN
4536      1A8A 1964
4537      1A8C 020A      LI   IOPARM, LFCR      PRINT LINE FEED/CARRIAGE RETU
4538      1A8E 0100
4539      1A90 0420      BLWP @PRINT
4540      1A92 19CE
4541      1A94 020A      LI   IOPARM, NAMMSG      PRINT PROGRAM TITLE

```

```

4532 1A96 1AE6'
4532 1A98 0420 BLWP @PRINT
4532 1A9A 19CE'
4533 *
4534 * PREDEFINED REGISTER OPTION
4535 REQPR
4536 1A9C 020A LI IOPARM, PDRMSG REQUEST IF PREDEFINED
4536 1A9E 1B02'
4537 1AA0 0420 BLWP @PRINTN REGISTERS DESIRED?
4537 1AA2 19D6'
4538 1AA4 020A LI IOPARM, INSBUF INPUT RESPONSE
4538 1AA6 017C'
4539 1AA8 0420 BLWP @KEYIN
4539 1AAA 198A'
4540 1AAC C820 MOV @SYMTAB, @SYMTBE INIT TOP OF SYMBOL TABLE
4540 1AAE 1154'
4540 1AB0 0006'
4541 1AB2 9820 CB @INSBUF, @NO NO PREDEFINED REGISTERS?
4541 1AB4 017C'
4541 1AB6 1B19'
4542 1AB8 1315 JEQ EXTMS THEN EXIT
4543 1ABA 9820 CB @INSBUF, @YES YES?
4543 1ABC 017C'
4543 1ABE 1B1A'
4544 1AC0 1303 JEQ SETPR
4545 1AC2 028A CI IOPARM, INSBUF ASSUME DEFAULTS IF NO INPUT
4545 1AC4 017C'
4546 1AC6 16EA JNE REQPR NO, ASK AGAIN
4547 SETPR
4548 1AC8 0209 LI R9, PDREGE-PDREG RESET NEXT SYMBOL TABLE ENTRY
4548 1ACA 0060
4549 1ACC 6809 S R9, @SYMTBE PAST PREDEFINED REGISTERS
4549 1ACE 0006'
4550 1AD0 0209 LI R9, PDREG MOVE PREDEFINED REGISTERS TO
4550 1AD2 1B1C'
4551 1AD4 020A LI R10, SYMT BOTTOM OF SYMBOL TABLE
4551 1AD6 1154'
4552 1AD8 022A AI R10, PDREG-PDREGE+1 TOP OF SYMBOL TABLE
4552 1ADA FFA1
4553 MOVPR
4554 1ADC DEB9 MOVB *R9+, *R10+ MOVE BYTES
4555 1ADE 0289 CI R9, PDREGE ALL DEFINITIONS MOVED?
4555 1AE0 1B7C'
4556 1AE2 16FC JNE MOVPR NO, CONTINUE
4557 EXTMS
4558 1AE4 045B RT START ASSEMBLY
4559 *
4560 * MESSAGES & DATA
4561 *
4562 1AE6 0A NAMMSG BYTE >0A
4563 1AE7 50 TEXT 'PXRASM 948925 ** 12JUL76'
4564 1B00 0A BYTE >0A, >0D
4564 1B01 0D
4565 1B02 50 PDRMSG TEXT 'PREDEFINED REGISTERS?'
4566 1B18 0D

```

4567	1B19	4E	NO	BYTE 'N'
4568	1B1A	59	YES	BYTE 'Y'

```
4570 *  
4571 * PREDEFINED REGISTER DEFINITIONS  
4572 *  
4573 EVEN  
4574 PDREG  
4575 1B1C 000F DATA 15 R15  
4576 1B1E 52 TEXT 'R15'  
4577 1B21 3B BYTE >3B  
4578 1B22 000E DATA 14 R14  
4579 1B24 52 TEXT 'R14'  
4580 1B27 3B BYTE >3B  
4581 1B28 000D DATA 13 R13  
4582 1B2A 52 TEXT 'R13'  
4583 1B2D 3B BYTE >3B  
4584 1B2E 000C DATA 12 R12  
4585 1B30 52 TEXT 'R12'  
4586 1B33 3B BYTE >3B  
4587 1B34 000B DATA 11 R11  
4588 1B36 52 TEXT 'R11'  
4589 1B39 3B BYTE >3B  
4590 1B3A 000A DATA 10 R10  
4591 1B3C 52 TEXT 'R10'  
4592 1B3F 3B BYTE >3B  
4593 1B40 0009 DATA 9 R9  
4594 1B42 52 TEXT 'R9'  
4595 1B44 003A DATA >3A  
4596 1B46 0008 DATA 8 R8  
4597 1B48 52 TEXT 'R8'  
4598 1B4A 003A DATA >3A  
4599 1B4C 0007 DATA 7 R7  
4600 1B4E 52 TEXT 'R7'  
4601 1B50 003A DATA >3A  
4602 1B52 0006 DATA 6 R6  
4603 1B54 52 TEXT 'R6'  
4604 1B56 003A DATA >3A  
4605 1B58 0005 DATA 5 R5  
4606 1B5A 52 TEXT 'R5'  
4607 1B5C 003A DATA >3A  
4608 1B5E 0004 DATA 4 R4  
4609 1B60 52 TEXT 'R4'  
4610 1B62 003A DATA >3A  
4611 1B64 0003 DATA 3 R3  
4612 1B66 52 TEXT 'R3'  
4613 1B68 003A DATA >3A  
4614 1B6A 0002 DATA 2 R2  
4615 1B6C 52 TEXT 'R2'  
4616 1B6E 003A DATA >3A  
4617 1B70 0001 DATA 1 R1  
4618 1B72 52 TEXT 'R1'  
4619 1B74 003A DATA >3A  
4620 1B76 0000 DATA 0 R0  
4621 1B78 52 TEXT 'R0'  
4622 1B7A 003A DATA >3A  
4623 1B7C PDREGE EQU *
```

PXRASH
INITIALIZATION

MIRA990 V2L1 17:44:10
948925-9901**

197/76

PAGE 0129

4624
4625

*

END START

R AAORG	0672	R AAORG1	0684	R AAORG2	0686	R AAORG3	0698
R AAORG4	06B4	R ABDAT	1754	ABDVL	0008	R ABES	06BA
ABET	0001	R ABOK	1734	R ABORT	0102	R ABRTFL	010C
R ABSIN	1478	R ABSS	06D4	ABVL	0003	R ABYTE	06E2
R ABYTE2	06F6	R ACHAR	0CD6	R ADAT2	0740	R ADAT3	075A
R ADAT4	075E	R ADATA	071A	R ADATA2	071E	R ADDUP	10D4
R ADEF	058C	R ADEF2	0778	R ADF	1014	R ADORG	0798
R ADORG1	07AA	R ADORG2	07AE	R ADORRD	14F4	R ADX2	07D8
R ADXOP	07C6	R AEND	0806	R AEQAB	0914	R AEQDF	091C
R AEQREL	090E	R AEQU	08E0	R AEQU2	08EE	R AEQU4	0904
R AER	10B0	R AEVEN	0B10	R AEVOUT	0B28	R AFDFBG	0E32
R AGET1	1308	R AIDT	0992	R AINST2	13F6	R AINSYM	077E
R AL	0CC2	R ALIST	09AC	R ALPHA	0C48	R AINDFER	0566
R ANF2	05AE	R ANODEF	056C	R ANTFND	059E	R ANXCHR	05D2
R ANXTFL	09E4	ADD	0002	R AOINST	13E8	R APAGE	09B2
R APERLN	0542	R APRLN	0542	R APROC	054A	R AR2	051E
R AREAD	050A	R AREF	0584	R AREL	0592	R ARF2	09D4
R ARFER	09EA	R ARORG	09F0	R ARORG2	09FE	R ARORG3	0A24
R ASCVL1	0BC8	R ASCVL2	0BCA	R ASM	0000	D ASMSCR	0448
ASTR	002A	R ATITL	0944	ATSGN	0040	R ATXT	0A4C
R AUNL	0AA8	R BATCH	0494	BDCODE	0008	R BGPRT	187A
BHCODE	0007	R BIHEX	0BE4	R BIN	0C04	R BINDC	0BF8
R BINDEC	0BF4	BINDPT	0007	R BINHEX	0BE0	BINVAL	0000
R BINVL	0BC0	BINVLP	0000	BINVLP	0004	R BLANK	17DC
BLNKK	0001	BLNKR	000A	BLNNK	000A	BLPT	0004
R BNXOP1	0F10	R BNXOP2	0F12	R BOPL0P	0EF4	BRANCH	000B
BUFADR	0005	BUFLN	0006	BVAL	0002	R BYTFIX	1650
BYTFL	0008	R C1LUNO	0060	R C2LUNO	0062	CDALIM	0005
CDLIM	0009	CGET	0001	CHAR	0002	CHOPC	000E
CHRCNT	0007	R CKCOM	176C	R CKENT	15F2	R CKER	007A
CKPT	000C	R CKSM	16E8	CKSUM	0001	R CLASS	01BA
R CLNBUF	17AE	R CLRBF	04C4	R CLRBG	17C2	R CLRFR	1612
R CLROBJ	17BE	R CLROP	060A	R CLRS	0502	CLSTYP	0001
R CNOD	183A	R COLON	0002	COMMA	000F	R COMPAR	00B2
R CONGET	0C5A	R CONLOC	1784	R CONPT	182A	R CONVMS	0BC0
COUNT	0001	R CR	0101	DBRN	000A	R DECM	1806
R DECM2	1836	DECP	0000	R DEF	0766	R DEFF	0768
R DFER	078A	R DIPR	063C	R DIVD	1102	R DOLLAR	0D26
R DORGFL	11C8	R DROUT	0E5A	R ED	101E	R EDF	006A
R EIGHT	0070	R ENDA	047C	R ENDACT	0460	R ENDAL	0812
R ENDEXT	08C8	R ENDFRT	11B6	R ENDIT	0EBE	R ENDLP	0890
R ENDMSG	010E	R ENDPRG	0064	R ENDRST	08D2	R ENDST	0B38
R ENDTRN	113A	R ENDV	085E	R ENO2	0DE0	EOF	2000
R EOFOD	0067	EOL	000D	R ER	105A	R ER2	0B4E
R ER3	0B52	R ER4	0B56	R ER5	0B5A	R ER6	0B5E
R ER7	0B62	R ER8	0B66	R ERCT	1074	R EREND	08D6
R EREX	10B6	R ERF	0390	R ERINS	108E	R ERL0C	001A
R ERMSG	0008	R ERNOEV	08DA	R ERNUM	0011	EROFST	0008
ERR4	0004	R ERREX	1586	ERRG	0000	R ERROR2	0B76
R ERROR3	0B74	R ERROR4	0B72	R ERROR5	0B70	R ERROR6	0B6E
R ERROR7	0B6C	R ERROR8	0B6A	R ERROR8	0CDA	R ERRS	1042
R ERVL	1046	R EVEN	093C	R EVLOC	0886	R EVN	180A
R EXBG	0F46	R EXCON	0F60	R EXE	160E	R EXEVL	0AAE
R EXIT2	152A	R EXIT4	154A	R EXITNO	0DCE	R EXLK	116C
R EXLP	0BBA	R EXPAD	14EA	R EXPR	0F42	EXPREG	0000

R EXPREX	0FA4	EXPRT	000E	R EXPRX2	104A	R EXT2	1520
R EXT3	1528	R EXTDFR	0E56	R EXTDFR	0E50	R EXTMS	1AE4
R EXT0	0A8C	R EXTQ2	0A92	R EXTRFA	0EC6	R EXTRFR	0EC0
R EXWP	0F1C	R EXWP2	0F22	FBYT	0004	R FF	0005
R FILM	10F6	R FIXFRF	15E6	FL	000A	FLAGS	0002
FLGREG	0005	FLGS	0004	R FOUND2	161E	R FR	102E
FR2	0006	R FRB	11DC	R FRBG	11D4	FRE	0001
R FRENT	11D0	R FREX	1052	R FRGT	104C	FRT	00A0
FRTCNT	0000	FRTEND	0009	FRTPT	0001	FRTVAL	0003
R FRWP	11B0	FST	0007	FSTIM	000E	R FSTPT	1858
R FTAB	1A50	FVL	0006	R FWDIN	1450	R FWDJ	1458
R FWDMK	15A4	FWDSN	0000	R FXFBG	15EA	R FXFRP	15C6
R GAD	14CC	R GCEXT	1A2A	R GCL00P	1A1C	R GDTRM	12D0
GETC	0001	R GETCHR	116E	R GETCNT	1A1A	R GETSCT	0D5C
R GETSVL	0D6E	R GO	0F9C	GSHLD2	000A	GSHOLD	0009
R GTCH	1172	R GT0	0A64	R GTSCT	0D66	R GTSCT2	0D60
R GTSFL	0D52	R GTSNM	0D84	R GTSVL	0D78	R GTSVL2	0D72
GVLPT	000F	HCNT	0009	R HEX	0CB0	R HEXX	0CB2
HPTR1	0006	HPTR1V	0007	R IDTFX	09A2	R ILEX	1054
R ILF	1038	R ILFRM	12CA	R IN0	142A	R IN1	1428
R IN10	1444	R IN11	1448	R IN12	144C	R IN2	142E
R IN3	1434	R IN4	1426	R IN5	142C	R IN6	1432
R IN7	1438	R IN8	143C	R IN9	1440	INCSC	0008
R INDRCT	1538	R INIT	1A6C	R INPT	18F4	R INSBUF	017C
INSFLG	0004	INSPT	0007	INSPT2	0000	R INST	1264
R INSTOP	11EE	INSTRG	0007	R INSTWD	1142	R INSVAL	0138
R INTBL	1A5E	R IOBUF	012A	IOC	0002	IOERR	4000
IOPARM	000A	IOPLUN	0003	R IOWKS	1944	R JMP1	0C4E
R JMP2	0CC0	R JMP3	0F78	R JMP6	1010	R JMPF	0E4E
R JMPIN	1424	R JMPTB1	1A2E	R JMPTB2	1A33	R JMPTB3	1A38
R JMPTB5	10CC	R JMPTB6	1A42	R JMPTB7	129A	R JMPTB8	064C
JVAL	000A	R KBLUNO	005C	R KEYIN	198A	R KEYPLF	19B8
R KEYRD	1992	R LABPTR	0068	R LABSN	15A6	LCKRG	0001
LEN	000A	R LERADD	001E	R LERLOC	0030	R LERMSG	0020
R LF	0100	R LFCR	0100	R LIMCHK	1192	R LIMEND	11AE
R LIST	1856	R LKC	1160	R LKCALL	118A	R LKCHAR	115C
R LKWP	113C	LMSGPT	0004	R LNCNT	1590	LNCTR	0001
R LNREG	184A	R LOCNUM	0132	LOCP2	000B	LOCP3	000C
R LPBG	0BA6	R LPCHR	117C	R LPOT	0E74	R LPOT2	0E88
R LPWP	0B2E	LSTPT	0006	MADD	0002	R MAINW	0428
R MLB	11FE	R MLBFX	11F2	R MLBG	11F6	MLBPT	0003
R MOVAL	0C02	R MOVCHR	0C56	R MOVPR	1ADC	R MOVWD	14FA
R MSGPRT	0B90	MSGPT	0003	R MULT	10E8	R MVCNT	0C76
R NAMMSG	1AE6	NCONS	0000	R NEWFRF	006C	NFA	0004
NFRT	0002	NFWD	000D	R NFNDA	11B8	NGFL	000F
R NO	1B19	R NOADD	0C6E	R NOADJ	061A	R NOEV	086C
R NOM	0A5A	R NONEG	10C8	R NOOPF	0622	R NOPST	0600
R NOTSAM	0DC2	R NOVFL	1670	R NPAGE	1592	R NO	0D08
R NQ2	0D12	NSBIT	0008	R NSMBL	0436	NSYM	0004
R NU	0CCC	NULL	0009	R NUM	0C8C	R NUMAD	0C9E
NUMERG	000B	R NUMERR	0B44	R NUMLOC	0044	R NUMMSG	0043
R NUMO	0C9A	NWSCT	0006	NWSYM	0007	R NWSYM2	0E3E
NWVAL	0000	R NWVL	0428	R NX1	0E1C	R NXCH2	0CF2
R NXCHAR	0CE4	NXCPT	0007	R NXCRD	061E	R NXGET	0C92
R NXOUT	0AE6	R NXSYM	0EA6	R NXSYMO	0EB0	R NXTFLD	0ACC

NXTWD	0003	NXWD	0003	R NXWLC	1480	R OBJBUF	0BD8
OBJPT	0004	OBJPT2	0007	R OBJRCD	0078	R OBJST	0085
R OBJSTP	0BD6	R OCHR	118C	R ODD	1818	R OGO	18E0
R OL2	16BA	OP	0008	R OPBG	1280	OPCNT	002F
OPCODE	0009	R OPEN	1964	R OPERX	0636	R OPEXT	13A8
R OPNCOD	0066	OPPTR	0007	OPRES	0005	R OPSBG	0EEE
R OPSRCH	0EEA	R OPSTK1	0FAA	R OPSTK2	0FB0	R OPSTK3	0FB8
R OPTBL	01FA	R OPTEND	0422	R OPTYPE	127C	R OPWP	1256
R OPWP2	125C	OPWRD	0008	R OQ	0D24	R OTBBG	17EC
R OTBG	170E	R OTBWP	17C8	R OTLN2	1466	OTOBPT	0009
R OTRCD	1764	R OTWP	16E6	R OTWRT	1794	R OUTB	17F4
R OUTBG	1220	R OUTBYT	17E8	R OUTEX	1254	R OUTNW	0B0E
R OUTNW2	0AE8	R OUTO	1716	R OUTOB	170A	R OUTOBJ	1706
R OUTRCD	1724	OUTS	000C	R OUTSYM	121C	R OVAL	1742
R OVR	18EC	R PAGNUM	00FB	PC	0004	PCNT	000A
R PCFRN	1490	R PCVL	0430	R PDREG	1B1C	R PDREGE	1B7C
R PDRMSG	1B02	R PGBG	15B2	PGCNT	0002	PGHC	000B
R PGHDR	15AE	PGP	000C	R PGWP	158E	R POS	162C
R PPC	11BA	PPCPT2	0008	PPCRG	0005	PRBF	0001
R PRBL2	18AC	R PRBLD	188C	R PRBUF	013E	R PRCR	18D4
R PRENT	19E8	R PREXT	1A18	R PRG	0794	R PRG2	06E0
R PRG3	0940	R PRG4	0A44	R PRINT	19CE	R PRINTN	19D6
R PRINTP	19D2	R PRLUNO	005E	PRNPT	000E	R PRNTLN	186A
R PROOP	05C4	R PRP2	1120	R PRP3	112A	R PRPMD	110E
R PRT	186E	R PRTAB	18B0	R PRTB1	18B2	R PRTCR	19DA
R PRTKEY	19F6	R PRTNCR	19E6	R PRTPCR	19DE	PRTPOS	0003
R PRTSP	18C0	R PRWRT	19EC	R PTLOC	1698	R PTLOC2	16A6
R PTLOC3	123C	PTR1	0001	PTR2	0002	R PXRASM	0450
R QT	1A6B	QTCLS	0000	R QUOTE	0D00	R0	0000
R1	0001	R10	000A	R11	000B	R12	000C
R13	000D	R14	000E	R15	000F	R2	0002
R3	0003	R4	0004	R5	0005	R6	0006
R7	0007	R8	0008	R9	0009	R RA	1552
RAPT	000D	RCD1	000C	R RCDCNT	1862	R RCDNUM	012C
R RD	190E	R RD2	1912	R RDBF	0002	R RDCOD	0900
RDCODP	0000	R RDF	1012	R RDRCD	190A	R RDYMSG	0119
R READ	1980	R REF	09C0	R REFDEF	0E2E	R REFF	09C2
R REG	152E	REGREG	000C	RELABS	0003	R RELEND	0874
R RELIN	1472	R RELNRK	013C	R RELLOC	1638	R RELOG	0A0A
R RELPC	0C18	RELREG	0004	RELVL	0006	R REQPR	1A9C
RESLT	0003	RESLT2	0004	R RFDFFX	081E	R RLDRG	07BE
RLVAL	0007	RMK	0005	R ROAB	006F	R ROOM	1728
R ROUT	0A28	R ROUT2	0A40	R RRG	09BC	RSV	0009
RSV3	0008	RSVAL	0000	RTN	000B	SABS	0002
SAVE	000C	SBYT	0003	R SCAL	0FF0	R SCAN	0C2C
R SCBG	0C30	R SCBLK	1948	SCCODE	0003	R SCHK	0E02
SCNR	0006	SCNT	0009	SCNT2	0007	SCPT	0005
SCPTPT	0000	R SCOT	0FC8	R SCOT2	0FEE	SCVAL	000A
R SCWP	0C0C	R SER	133A	R SERRG	13D2	R SETOUT	168E
R SETPR	1AC8	SFLAGS	000A	SHLD	0006	R SINFX	10F8
SIXTN	0010	SNAM	0006	SP	0009	SPTR	0008
R SRC	0D98	R SRCSYM	0D94	R START	0448	STEP	0004
R STK	0FBE	R SUB	10E0	SVAL	000C	X SVC	000F
E SVCSR	1A76	E SVCWP	1A6E	SVN	0002	SYM	0005
R SYMBL	114A	R SYMBOL	0D32	R SYMCBG	0DFA	R SYMCHK	0DA0

R SYMCLR	0DF6	R SYMLOC	0E6A	SYMNM	0003	SYMNO	0001
SYMNPT	000D	SYMPTR	0000	E SYMT	1AD6	R SYMTAB	1154
R SYMTBE	0006	R SYNRFX	05FC	R SYNTAX	0790	R SYNTAX2	0A48
R TAB	0004	TABCHR	0006	R TABS	1598	TCHR	0002
TEMP	0000	TEMP2	000A	TEMPIN	0006	TEMPRG	0008
R TENS	0072	R TERM	0001	THOLD	0008	R TITFX	00C3
R TITFX2	094C	R TITLE	00C2	TITPT	0003	R TITQ	0966
R TITQ2	098C	R TITQ3	098C	R TPAGE	00F5	TQC	000A
R TQUOT	0D18	R TRERR	168A	R TRN	10DA	R TRNEX	10DE
R TRUNK	112C	R TWO	11BC	R TXTEND	0AA4	R TY72	13AE
R TYP0	1348	R TYP1	12F0	R TYP2	1316	R TYP3	1322
R TYP4	134A	R TYP4A	1362	R TYP5	136E	R TYP6	1374
R TYP7	1386	R TYP8	1354	R TYP9	13D8	R TYPB	131C
R TYPC	12B4	TYPE	000C	R UNDEF	1234	R UNLOC	0056
R UNLFP	0ED6	R UNMSG	0036	R UNDSM	0ECC	R UNLMSG	0051
R VALCHK	157C	R VALCKP	1578	R VALOP	10B8	R VALOP2	10BC
R VALOUT	158C	VALSV	0001	VALUE	0005	R VCK	12DE
R VCKP	12DA	R VL	0D3C	R VOK	12EC	R WDE	0AC0
WORD1	0000	WORD2	0002	WRCOD	0B00	WRCODP	0001
R WREOF	192E	R WRITE	19C2	R XOP2	14C4	R XOPBG	149E
R XOPG	149A	R YES	1B1A	R Z1	006E	ZCK	0005
ZCK2	000C	R ZT	0952	R ZZ1	0B40	ZZERO	0005

0000 ERRORS

AAORG	0918	0895			
AAORG1	0925	0923			
AAORG2	0926	0921			
AAORG3	0931	0929	1065		
AAORG4	0941	0934			
ABDAT	3884	3882			
ABDVL	0217	3880			
ABES	0947	0896			
ABET	0224	3997	4034		
ABOK	3867	3862			
ABORT	0380	2845			
ABRTFL	0382	0706	1164	2847	
ABSIN	3353	3299			
ABSS	0959	0897			
ABVL	0212	3860			
ABYTE	0968	0898	0986		
ABYTE2	0975	0973			
ACHAR	1792	1790			
ADAT2	1007	0996			
ADAT3	1016	1014			
ADAT4	1018	1006			
ADATA	0991	0899			
ADATA2	0993	1021			
ADDUP	2645	2638			
ADEF	0786	0784			
ADEF2	1033	1040	1043		
ADF	2545	4439	4440	4443	4444
ADORG	1051	0901			
ADORG1	1058	1056			
ADORG2	1060	1054			
ADGRPD	3473	3470			
ADX2	1079	1076			
ADXOP	1072	0902			
AREND	1098	0903			
AREAB	1196	1193			
AREDF	1200	1198			
AREREL	1194	1191			
AREOU	1176	0904			
AREOU2	1181	1179			
AREOU4	1189	1187			
AREP	2618	2616			
AREVEN	1447	0842	0992	1217	
AREVOUT	1456	1451			
AREVFB6	2153	2152			
ARET1	3148	3137	3159	3223	3415
ARET	1260	0906			
ARENST2	3286	3091	3166	3261	3264 3283
ARENSYM	1036	1031			
ARE	1793	4417			
ARE1ST	1273	0907			
AREPHA	1719	4411			
AREPFR	0771	0776			
AREP	0801	0794	0799		
AREPFR	0774	0770			
AREPFR	0795	0767			

XREF990	VILL1	09:30	25	198/76	PAGE				0136		948925-9901		**
BYTFL	0231	0719	0719	0740	0740	1457	1457	3848	3848	4014			
		4048											
C1LUND	0330	0691	4264										
C2LUND	0331	0693	4201	4324									
CDALIM	0214	3855											
CDLIM	0218	3875											
CGET	0165	3138	3195	3224	3271	3281	3410	3462	3485	3488			
		3512	3521										
CHAR	0051	0751	0762	0816	0822	0832	1002	1003	1004	1086			
		1112	1234	1250	1322	1352	1357	1370	1382	1416			
		1418	1565	1567	1569	1726	1727	1754	1764	1766			
		1768	1784	1786	1788	1793	1804	1805	1806	1808			
		1810	1824	1831	1839	2437	2509	2592	2594	2596			
		2785	2786	2786	2787	2788	2790	2798	2803	2808			
		2811	2819	3282	3284	3411	3489						
CHOPC	0090	1621	1625	1636									
CHRCNT	0261	4293	4296	4367	4396	4398	4401	4404					
CKCOM	3902	3908											
CKENT	3660	3672	3785										
CKER	2446	2525	4423										
CKPT	0219	3861	3897	3904	3907								
CKSM	3792	3912											
CKSUM	0210	3898	3906	3909									
CLASS	0422	1812	2439										
CLNBUF	3930	3921	3933										
CLRBF	0713	0716											
CLRBG	3938	3937											
CLRFRT	3680	3683											
CLROBJ	3936	1106	1151	2208	2226								
CLROP	0839	0833											
CLRS	0729	0732											
CLSTYP	0095	1712	1714	1730	1732	1758	1779	1781	1812	1813			
		1820	1826	1836	2439	2440	2441	2495	2496				
CNOD	4042	4028	4038										
COLON	0291	1152											
COMMA	0182	3139	3196	3225	3272	3481							
COMPAR	2030	2034											
CONGET	1728	1735											
CONLOC	3913	3910											
CONPT	4032	3995											
CONVMS	1580	1618	1630										
COUNT	0110	0793	0793	1203	1203	1956	1957	1981	2202	2247			
CR	0375	2250	3916	4120	4154	4169	4192	4399					
D!		0001											
DBRN	0060	0890	0891	0892	0892	0893	1223	1226	1247	1267			
DECM	4002	3998											
DECM2	4039	4035											
DECPT	0186	3597											
DEF	1025	0900											
DEFF	1027	1035											
DEFP	1041	1029											
DIPP	0885	3101											
DIVD	2634	2641											
DOUBLE	1845	4413											
DORSEI	2859	0707	0922	0926	1053	1059	1313	1319	2608	2916			

NEWFRF	0356	0720	0738	0781	1100	2127					
NFA	0154	2924									
NFRT	0152	2933	2973								
NFWD	0134	2426	2488	2564	2564	2629					
NFWDA	2861	1407	2447	2562	2568	3171					
NGFL	0136	2422	2482	2631	2631	2633	2676	2676	2679	2700	
		2705									
NO	4567	4541									
NOADD	1737	1731	1733								
NOADJ	0844	0841									
NOEV	1133	1113	1172								
NOM	1356	1353									
NONEG	2635	2632									
NOOPF	0848	0828	0830								
NOPST	0836	0856									
NOTSAM	2042	2029	2032								
NOVFL	3734	3709	3711								
NPAGE	3560	0704									
NO	1823	1821									
NO2	1829	1827									
NSBIT	0117	2014	2046	2106	2122	2176	2179	2185	2233	2236	
		2243	2244	2246	2249	2250	2928	2928	3665	3665	
NSMBL	0631	2024	2056								
NSYM	0113	2024	2031	2056	2058						
NU	1787	4418									
NULL	0058	1397	1402								
NUM	1753	4412									
NUMAD	1763	1759									
NUMERG	0076	1562									
NUMERR	1475	0702	1154	1159	3020						
NURLOC	0317	1156									
NUMMSG	0316	1157									
NUMO	1760	1780	1785	4419	4420	4421					
NWSCT	0115	0796	0796	1080	1080	1297	1297	1745	1745	2028	
		2052	2054	2158	2205	2209	2223				
NWSYM	0056	0935	0936	0937	1224	1228	1242	1246	1268	1334	
		1335	1336	1747	1747						
NWSYM2	2160	2218									
NWVAL	0049	0709	0710	0711	0712	0777	0780	0785	0789	0791	
		0793	0796	0797	0800	0803	0804	0971	0972	0972	
		0976	1140	1147	1192	1188	1192	1199	1201	1203	
		1204	1205	1226	1227	1231	1240	1242	1243	1244	
		1244	1246	1297	1298	1300	1301	1363	1364	1365	
		1365	1377								
NWVL	0624	0978	1431								
NXJ	2121	2115									
NXCHR	1809	1807									
NXCHAR	1803	1658									
NXPRT	0099	1711	1729	1757	1778	1819	1825	1835			
NXCRD	0846	0854									
NXGET	1756	1769	1771								
NXOUT	1424	1419									
NXSUM	2214	2206	2252	4453	4455	4457	4459	4460			
NXSUM2	2322	2165									
NXFLD	1414	0985	1028	1034	1303						

PRPMD	2693	2665	2685							
PRT	4097	4096								
PRTAB	4135	4119								
PRTB1	4137	4142								
PRTCR	4349	4346								
PRTKEY	4365	4307								
PRTNCR	4356	4348								
PRTPCR	4352	4347								
PRTPOS	0243	4113	4130	4138	4146	4147				
PRTSP	4143	4139	4148							
PRMRT	4360	4326	4355							
PTLOC	3757	3755								
PTLOC2	3762	3759								
PTLOC3	3016	3014								
PTR1	0195	3654	3655	3656	3661	3663	3667	3668	3669	3670
		3671	3678	3684	3685	3692	3702	3759	3760	3779
		3783	3784							
PTR2	0197	3656	3667	3668	3669	3671	3677	3678	3681	3682
		3721	3732	3754	3755	3784				
PXRASM	0675	4205	4377							
QT	4484	1015	1195	3352	3767	3881				
QTCLS	0094	1820	1826	1836						
QUOTE	1818	1709	4414							
R0	0266	2731	2732	2733	2734					
R1	0267									
R10	0276	2731	2734	4114	4122	4123	4126	4127	4144	4145
		4149	4238	4238	4289	4289	4363	4363	4551	4552
		4554								
R11	0277	2608	2916	2964	3844					
R12	0278	4237	4238	4239	4239	4240	4288	4289	4290	4296
		4300	4301	4362	4363	4364				
R13	0279	4237	4288	4362						
R14	0280									
R15	0281	4372								
R2	0268									
R3	0269									
R4	0270									
R5	0271									
R6	0272									
R7	0273									
R8	0274									
R9	0275	4117	4118	4120	4122	4124	4136	4138	4140	4141
		4147	4189	4519	4520	4521	4522	4548	4549	4550
		4554	4555							
RA	3527	1089	3041							
RAPT	0180	3193	3216	3227	3269	3274	3486	3509	3513	
RCD1	0250	4106	4193							
RCDCNT	4068	0727	1549	1559						
RCDNUM	0398	4168	4170	4171						
RD	4178	4177								
RD7	4181	4184								
RDRF	0242	0748	0748	4117	4156	4179				
RDCDF	0345	4212								
RDCDDP	0254	1387								
RDF	2540	4437	4438	4441	4442					

RDRCD	4176	0749								
RDYMSG	0391	3922								
READ	4262	4188								
REF	1287	0909								
REFDEF	2150	1109								
REFF	1289	1304								
REG	3506	3458								
REGREG	0179	1093	1093	3144	3145	3153	3194	3217	3228	3229
		3270	3275	3276	3300	3313	3321	3324	3327	3455
		3543	3549							
RELABS	0052	0726	0787	0787	0798	0798	0920	0920	0925	1055
		1055	1064	1067	1134	1134	1315	1315	1317	1847
		1847	2605	2605	2615	2615	2977	2977	3258	3258
		3770	3770	3861	3861					
RELEND	1137	1135								
RELIN	3351	3300								
RELMRK	0403	1015	1195	3352	3764	3767	3881	4061	4168	4169
RELOC	3701	3699								
RELOGR	1320	1316								
RELPC	1657	0725	0924	1057	1136	1318				
RELREG	0199	3691	3694	3700	3735	3738	3740	3741	3746	3765
		3765								
RELVL	0127	0928	0928	1013	1013	1062	1062	1190	1190	1325
		1325	1399	1399	2429	2453	2453	2455	2574	2601
		2602	2646	2657	2694	2694	2975	2975	3210	3210
		3258	3258	3475	3475					
REQPR	4535	4546								
RESLT	0096	0826	0826	0932	0932	0971	0971	1009	1009	1208
		1208	1210	1210	1240	1240	1330	1330	1363	1363
		1401	1401	1710	1720	1721	1722	1723	1727	1746
		1747	1754	1755	1765	1766	1767	1768	1776	1789
		1793	1798	1824	1830	1831	1846	2427	2512	2512
		2573	2600	2627	2627	2647	2658	2666	2667	2667
		2671	2678	2686	2687	2688	2698	2701	2974	2974
		3123	3123	3174	3174	3208	3208	3251	3251	3477
		3477	3541	3541						
RESLT2	0125	2428	2671	2686						
RFDFFX	1107	1105								
RLDORG	1066	1063								
RLVAL	0128	1847	1847	2435	2541	2646	2657	2696	2696	
RMK	0245	4161								
ROAB	0358	0937	1335							
ROOM	3859	3856								
ROUT	1331	1323	1328							
ROUT2	1340	1333								
FRG	1283	1256	1265	1275	1281					
RSV	0177	3449	3495	3499	3510	3518	3523			
RSV3	0176	3149	3154							
RSVAL	0124	2512	2513	2514	2514	2516	2547	2554	2555	2556
		2557	2627	2634	2647	2658	2666	2688	2703	2706
RTN	0103	1834	4405							
SABS	0196	3698								
SAVE	0104	1834	1841							
SEVT	0226	4009	4010	4012	4020	4022	4027	4045	4046	4047
SCAL	2522	4428	4429	4430	4432					

TYP7	3234	3098								
TYP8	3201	3099								
TYP9	3268	3100								
TYPB	3164	3102								
TYPC	3107	3103								
TYPE	0062	1038	1039	1080	1081	1082	1084	1185	1186	1197
		1434	1443							
UNDEF	3013	3023								
UNDFOC	0321	3017								
UNDLP	2245	2248								
UNDM5G	0310	3011								
UNDSM	2041	4461	4462	4463						
UNLMSG	0320	3018								
VALCHK	3542	3539								
VALCKP	3540	3529								
VALDP	2626	2528								
VALDP2	2628	2518	2548							
VALOUT	3550	3546								
VALSV	0083	1626	1637	1641						
VALUE	0114	1130	1130	1972	2112	2547	2547	2600	2600	3120
		3120	3405	3405	3538	3538				
VCK	3124	3121								
VCKP	3122	3109								
VL	1860	2188								
VOK	3130	3128								
WDE	1406	0927	0970	1061	1177	1324	1398			
WORD1	0352	2334								
WORD2	0353	2336								
WORD3	0346	4213								
WORDF	0255	4304	4361							
WROF	4198	1168	4190							
WRITE	4322	1153	3920							
XOP2	3414	3412								
XOPB6	3398	3397								
XOPG	3395	0853								
YFS	4568	4543								
ZI	0357	1126	3883							
ZTK	0054	0738	0740	0827	0922	0928	0933	1013	1053	1062
		1100	1159	1164	1178	1190	1280	1313	1325	1332
		1399	1421	1439	1450					
ZCX2	0133	2451	2562	2603	2605	2615				
ZT	1239	1232								
ZTI	1473	1322								
ZTFR0	0228	0709	0709	1138	1138	4010	4021	4022		

THERE ARE 0749 SYMBOLS

0003 *
0004 * TITLE: PXRLAL
0005 * LINKING LOADER
0006 * REVISION: 05/01/74
0007 * ORIGINAL
0008 * 03/15/76
0009 * MODIFIED TO BE OVERLAY COMMAND OF PXRMT
0010 * 06/28/76
0011 * PULLED FROM OLD MONITOR FOR USE IN
0012 * CORESIDENT ED/ASM/TRC MONITOR.
0013 * MODIFIED TO BE COMPATIBLE WITH EXTENDED MONITOR
0014 * (INCLUDES ASSEMBLER AND TEXT EDITOR)
0015 * ALSO MODIFIED SO IT SETS UP XOP 15 TRAP.
0016 * COMPUTER: 990,990 ASSEMBLY
0017 * ABSTRACT: PXRLAL IS A VERSION OF LAL990 MODIFIED
0018 * TO RUN WITH PXRMT. THE PROTOTYPING SYSTEM
0019 * AND 733ASR PROGRAM DEVELOPMENT SYSTEM DEBUG
0020 * MONITOR
0021 * PXRLAL LOADS OBJECT MODULES INTO MEMORY,
0022 * PERFORMS THE LINKING DEFINED IN THE
0023 * PROGRAM MODULES, PERFORMS THE ADDRESS
0024 * MODIFICATION FOR RELOCATABLE CODE, AND
0025 * PRINTS A LOAD MAP.
0026 *

```
0029 * TITLE: DATBS
0030 * DATA BASE
0031 * REVISION: 05/01/74
0032 * ORIGINAL
0033 * 03/15/76
0034 * MODIFIED TO RUN WITH PXRMTN
0035 * COMPUTER: 990, 990 ASSEMBLY
0036 * ABSTRACT: THIS DATA BASE CONTAINS ALL WORKSPACE
0037 * REGISTER EQUATES, ERROR MESSAGES, AND
0038 * MISCELLANEOUS DATA
0039 * CALLING SEQUENCE: NON-CALLABLE
0040 *
0041 IDT 'PXRLAL'
0042 *
0043 * EXTERNAL REFS AND DEFS
0044 *
0045 REF ERROR, USRPC, INIT, LDBUF, GETHEX, PROCRLF, SVCALT
0046 REF SVCWP, SVCSR
0047 DEF BEGLAL
0048 DEF ENDLAL
0049 DEF LALCSR
0050 *
0051 * MAINWP REGISTERS
0052 *
0053 0000 TEMP EQU 0
0054 0001 RDRBF EQU 1
0055 0002 ESTPT EQU 2
0056 0003 SYMP EQU 3
0057 0004 ENTPT EQU 4
0058 0005 ENTVAL EQU 5
0059 0006 PRNTN EQU 6
0060 0007 SPTR EQU 7
0061 0008 TPPT EQU 8
0062 0009 UNDPT EQU 9
0063 000A IOPARM EQU 10
0064 000C SFLAGS EQU 12
0065 *
0066 * DSMP REGISTERS
0067 *
0068 0000 INPT EQU 0
0069 0001 TGFLD EQU 1
0070 0002 BIAS EQU 2
0071 0003 NEWLOC EQU 3
0072 0004 LOCPT EQU 4
0073 0005 SVPT EQU 5
0074 0006 HLD EQU 6
0075 0007 FLGVL EQU 7
0076 0008 PC EQU 8
0077 000C LDPT EQU 12
0078 000D INCP EQU 13
0079 000E INSID EQU 14
0080 000F TDAT EQU 15
0081 *
0082 * LDWP REGISTERS
0083 *
0084 0000 RD EQU 0
0085 0001 FST EQU 1
0086 0002 PTR EQU 2
0087 0003 NSMBLP EQU 3
0088 0004 SYMB EQU 4
```

```

0089      0005  TMP      EQU    5
0090      0006  LOC      EQU    6
0091      0007  SRSM     EQU    7
0092      0008  NMMSG    EQU    8
0093      000C  HEXP2    EQU   12
0094      *
0095      *          CMWP  REGISTERS
0096      *
0097      0000  CKL      EQU    0
0098      0001  HVAL     EQU    1
0099      0002  RDRL     EQU    2
0100      0003  CUMPT    EQU    3
0101      0004  ENDOB    EQU    4
0102      0005  HEXP     EQU    5
0103      0006  RLRG     EQU    6
0104      0007  CSUM     EQU    7
0105      *
0106      *          CONVWS REGISTERS
0107      *
0108      0000  BINVAL   EQU    0
0109      0001  VALSV    EQU    1
0110      0002  MADD     EQU    2
0111      0003  SCCODE   EQU    3
0112      0007  BHCODE   EQU    7
0113      0008  HBCODE   EQU    8
0114      0009  SVCAL1   EQU    9
0115      000B  HCNT     EQU   11
0116      000C  PCNT     EQU   12
0117      000E  CHOPC    EQU   14
0118      *
0119      *          SYMBOL REGISTERS
0120      *
0121      0000  SYMPTR   EQU    0
0122      0001  SYMNO    EQU    1
0123      0002  FLAGS    EQU    2
0124      0003  SYMM     EQU    3
0125      0004  NSYM     EQU    4
0126      0005  VALUE    EQU    5
0127      0006  SEVEN    EQU    6
0128      0007  STPTR    EQU    7
0129      0008  NWSCT    EQU    8
0130      0009  TN       EQU    9
0131      000A  NSBIT    EQU   10
0132      000C  NINE     EQU   12
0133      *
0134      *          I/O  WORKSPACE EQUATES
0135      *
0136      0000  RDCODP   EQU    0
0137      0001  WRCDP    EQU    1
0138      0002  IOC      EQU    2
0139      0003  IOPLUN   EQU    3
0140      0004  FLGS     EQU    4
0141      0005  BUFADR   EQU    5
0142      0006  BUFLN    EQU    6
0143      0007  CHRNT    EQU    7
0144      0008  LEN      EQU    8
0145      0009  SVCAL2   EQU    9
0146      000B  RTN      EQU   11
0147      *
0148      *          COMMON REGISTER EQUATES

```

0149		*		
0150	0000	R0	EQU	0
0151	0001	R1	EQU	1
0152	0002	R2	EQU	2
0153	0003	R3	EQU	3
0154	0004	R4	EQU	4
0155	0005	R5	EQU	5
0156	0006	R6	EQU	6
0157	0007	R7	EQU	7
0158	0008	R8	EQU	8
0159	0009	R9	EQU	9
0160	000A	R10	EQU	10
0161	000B	R11	EQU	11
0162	000C	R12	EQU	12
0163	000D	R13	EQU	13
0164	000E	R14	EQU	14
0165	000F	R15	EQU	15
0166		*		
0167		*	ERROR CODES	
0168		*		
0169	2301	ILSQ	EQU	>2301
0170	2302	ILCD	EQU	>2302
0171	2303	MSEND	EQU	>2303
0172	2304	LAER	EQU	>2304
0173	2305	PRLM	EQU	>2305
0174	2306	CKSM	EQU	>2306

ILLEGAL LOAD SEQUENCE
ILLEGAL LOAD CODE
MISSING END STATEMENT
LOAD ADDRESS ERROR
PREVIOUS LOAD MODULE
CHECKSUM ERROR

```

0176 *
0177 * MONITOR COMMAND OVERLAY SECTION
0178 *
0179 0000 4C BEGLAL TEXT 'LL'
      0001 4C
0180 0002 00BE DATA LALCSR
0181 0004 0000 DATA 0
0182 *
0183 * DATA
0184 *
0185 0006 0A0D LFCR DATA >0A0D LINE FEED/CARRIAGE RETURN
0186 0006 LF EQU LFCR
0187 0007 CR EQU LFCR+1
0188 0008 20 BLANK BYTE >20
0189 2000 EOF EQU >2000 END OF FILE STATUS
0190 4000 IOERR EQU >4000 UNRECOVERABLE ERROR STATUS
0191 0009 3A COLON BYTE ':'
0192 000A 46 F BYTE 'F'
0193 000B 4C LOAD BYTE 'L'
0194 000C 45 END BYTE 'E'
0195 000D 54 TERM BYTE 'T'
0196 *
0197 * MONITOR SUPV CALL CODES FOR I/O
0198 *
0199 0900 RDCOD EQU >0900 READ ASCII
0200 0B00 WRCOD EQU >0B00 WRITE ASCII
0201 000E EVEN
0202 *
0203 * PERIPHERAL DEVICE DEFAULT LOGICAL UNIT NUMBERS
0204 *
0205 000E 0000 KBLUNO DATA 0 LOG (KEYBOARD)
0206 0010 0006 PRLUNO DATA 6 LOG (PRINTER)
0207 0012 0007 CTLUNO DATA 7 CS# (DEFAULT CASSETTE LUNO)
0208 *
0209 * DEFAULT LOAD VALUES
0210 *
0211 0000 DLDPT EQU 0 LOAD POINT
0212 00A0 DLDBI EQU >A0 LOAD BIAS
0213 *
0214 * DEFINITION MESSAGE
0215 *
0216 0014 20 DEFM TEXT ' *
      0015 20
      0016 20
      0017 20
      0018 20
      0019 20
      001A 20
      001B 20
      001C 20
      001D 20
      001E 20
      001F 20
      0020 2A
      0021 20
0217 0022 0000 DEFN DATA 0.0.0
      0024 0000
      0026 0000
0218 0028 20 TEXT '
      0029 20

```

```
0219 002A 0000 DEFL DATA 0.0
      002C 0000
0220 002E 00    BYTE >D
0221                *
0222                *   UNDEFINED SYMBOLS MESSAGE
0223                *
0224 002F 55 UNDEF TEXT 'UNDEF'           UNDEFINED SYMBOL
      0030 4E
      0031 44
      0032 45
      0033 46
0225 0034 0A    BYTE >A,>D
      0035 0D
0226 0036 20 UDS TEXT ' '
      0037 20
      0038 20
      0039 20
0227 003A 0000 UDSN DATA 0.0.0
      003C 0000
      003E 0000
0228 0040 0A    BYTE >A,>D
      0041 0D
0229                *
0230                *   MULTIPLY DEFINED SYMBOL MESSAGE
0231                *
0232 0042 20 MULDF TEXT 'M'           MULTIPLE DEFINED SYMBOL
      0043 4D
0233 0044 0D    BYTE >D
0234                *
0235                *   LOAD MESSAGE
0236                *
0237 0045 4C LODMSG TEXT 'LOAD/END? '
      0046 4F
      0047 41
      0048 44
      0049 2F
      004A 45
      004B 4E
      004C 44
      004D 3F
      004E 20
0238 004F 0D    BYTE >D
0239                *
0240                *   NAME MESSAGE
0241                *
0242 0050 20 NAMMSG TEXT ' '
      0051 20
      0052 20
      0053 20
      0054 20
      0055 20
0243 0056 0000 NAME DATA 0.0.0, ' '
      0058 0000
      005A 0000
      005C 2020
0244 005E 0000 LOCVAL DATA 0.0
      0060 0000
0245 0062 0D    BYTE >D
0246                *
0247                *   ENTRY MESSAGE
```



```
0248 *
0249 0063 20 ENTMSG TEXT / ENTRY = /
      0064 45
      0065 4E
      0066 54
      0067 52
      0068 59
      0069 20
      006A 3D
      006B 20
0250 006C LOCNUM BSS 4
0251 0070 0A BYTE >A.>D
      0071 0D
0252 *
0253 * END ACTION MESSAGE
0254 *
0255 0072 0A ENDACT BYTE >A
0256 0073 54 TEXT /TERM/CONT? /
      0074 45
      0075 52
      0076 4D
      0077 2F
      0078 43
      0079 4F
      007A 4E
      007B 54
      007C 3F
      007D 20
0257 007E 0D BYTE >D
0258 *
0259 * LIST MESSAGE
0260 *
0261 007F 0A LSTMSG BYTE >A
0262 0080 46 TEXT /F/P LIST? /
      0081 2F
      0082 50
      0083 20
      0084 4C
      0085 49
      0086 53
      0087 54
      0088 3F
      0089 20
0263 008A 0D BYTE >D
0264 *
0265 * REQUEST LOAD POINT MESSAGE
0266 *
0267 008B 0A LPMSG BYTE >A
0268 008C 4C TEXT /LD PT? /
      008D 44
      008E 20
      008F 50
      0090 54
      0091 3F
      0092 20
0269 0093 0D BYTE >D
0270 *
0271 * REQUEST LOAD BIAS MESSAGE
0272 *
0273 0094 0A LBMSG BYTE >A
```

DATA BASE

948926-9901

**

PAGE 0000

0274	0095	40
	0096	44
	0097	20
	0098	42
	0099	49
	009A	3F
	009B	20
0275	009C	00
0276	009E	

TEXT <LD BI?>

BYTE >D
EVEN

```
0279          * MAIN DRIVER WORKSPACE AREA
0280          *
0281 009E      MAINWP
0282 009E 0000 DATA 0          R0    TEMP
0283 00A0 0000 DATA LDBUF       R1    RDRBF
0284 00A2 059C DATA ENDST        R2    ESTPT
0285 00A4 0588 DATA SYMBOL+SYMPTR+SYMPTR R3    SYMP
0286 00A6 0063 DATA ENTMSG        R4    ENTPT
0287 00A8      ENTADD
0288 00A8 0000 ENTVL  DATA 0          R5    ENTVAL
0289 00AA 066C DATA PRINTN       R6    PRNTN
0290          REF ENDBUF
0291 00AC 0000 SYMTAB DATA ENDBUF       R7    SPTR
0292 00AE 0000 TOPDAT DATA 0          R8    TPPT
0293 00B0 0000 DATA 0          R9    UNDPT & I/O RTN STATUS
0294 00B2 0000 DATA 0          R10   IOPARM
0295 00B4 0000 DATA 0          R11   RETURN
0296 00B6 058C DATA SYMBOL+FLAGS+FLAGS R12  SFLAGS
0297 00B8 0000 DATA 0          R13
0298 00BA 0000 DATA 0          R14
0299 00BC 0000 DATA 0          R15
```

```

0301 * TITLE: PXRLAL
0302 * LINK AND LOAD DRIVER
0303 * REVISION: 05/01/74
0304 * ORIGINAL
0305 * 03/15/76
0306 * MODIFIED TO BE A COMMAND OF PXRMTX
0307 * COMPUTER: 990, ASSEMBLY
0308 * ABSTRACT: THE LIST OPTION IS REQUESTED. THE USER
0309 * SHOULD ENTER 'F' FOR FULL AND 'P' FOR
0310 * PARTIAL LIST. THE QUESTION 'LOAD/END?'
0311 * IS ASKED. THE USER SHOULD ENTER 'L' FOR
0312 * LOAD OR 'E' FOR END. IF LOAD, THE DISPATCH
0313 * ROUTINE IS CALLED TO PROCESS THE MODULE.
0314 * THE LOAD/END LOOP IS CONTINUED UNTIL
0315 * ALL MODULES HAVE BEEN LOADED.
0316 * IF END, THE ENTRY POINT AND ANY UNDEFINED
0317 * SYMBOLS ARE PRINTED, AND THE TERM/ CONT
0318 * QUESTION IS ASKED. IF THERE ARE STILL
0319 * SOME UNDEFINED SYMBOLS, THE CONT OPTION
0320 * MAY BE SPECIFIED AND ADDITIONAL MODULES
0321 * MAY BE LOADED. ONCE ALL MODULES ARE LOADED,
0322 * TERM MAY BE SPECIFIED, AND CONTROL WILL
0323 * RETURN TO THE MONITOR. THE USER MAY THEN
0324 * EXECUTE OR DEBUG HIS PROGRAM WITH MONITOR
0325 * COMMANDS.
0326 * CALLING SEQUENCE:
0327 * BL @LALCSR
0328 * STATISTICS: WORKSPACE = MAINWP (UNSHARED)
0329 * ROUTINES CALLED:
0330 * PRINT, PRINTN, KEYIN, DSPTCH, GETSFL,
0331 * GETSNM, BINHEX, PRCLRF, GETHEX, OPEN
0332 *
0333 * PXRLAL COMMAND INTERFACE
0334 *
0335 00BE LALCSR
0336 00BE 0420 BLWP @PXRLAL START UP LAL
0337 00C0 00C4 RT RETURN TO MONITOR
0338 *
0339 00C4 PXRLAL
0340 00C4 009E DATA MAINWP, LAL WORKSPACE, START
0341 00C6 00C8 LAL
0342 00C8 02E0 LWPI MAINWP INIT WP
0343 00CA 009E
0343 * SET UP XOP LEVEL 15 TRAP*****
0344 00C0 0209 LI R9, SVCWP
0345 00CE 0000
0345 00D0 C809 MOV R9, @>7C
0346 00D2 007C
0346 00D4 0209 LI R9, SVCSR
0347 00D6 0000
0347 00D8 C809 MOV R9, @>7E
0348 *****
0349 00DC 06A0 BL @PRCLRF PRINT CR/LF
0350 00DE 0000
0350 00E0 1000 NOP IGNORE ERROR RETURN
0351 00E2 RQLP
0352 00E2 020A LI IOPARM, LPMSG REQUEST LOAD POINT
    
```

```

00E4 008B'
0353 00E6 0416      BLWP *PRNTN
0354 00E8 06A0      BL   @GETHEX      INPUT HEX VALUE
      00EA 0000
0355 00EC 10FA      JMP  RQLP        IF ERROR
0356 00EE 1002      JMP  STLP        GOOD INPUT VALUE
0357 00F0 020A      LI   R10,DLDP   NO INPUT, ASSUME DEFAULT
      00F2 0000
0358 00F4          STLP
0359 00F4 C80A      MOV  R10,@LOADPT SAVE LOAD POINT
      00F6 01FE'
0360 00F8          RQLB
0361 00F8 020A      LI   IOPARM,LBMSG REQUEST LOAD BIAS
      00FA 0094'
0362 00FC 0416      BLWP *PRNTN
0363 00FE 06A0      BL   @GETHEX      INPUT HEX VALUE
      0100 00EA'
0364 0102 10FA      JMP  RQLB        IF ERROR
0365 0104 1002      JMP  STLB        GOOD INPUT VALUE
0366 0106 020A      LI   R10,DLDBI  NO INPUT, ASSUME DEFAULTS
      0108 00A0
0367 010A          STLB
0368 010A C80A      MOV  R10,@BIASWD SAVE LOAD BIAS
      010C 01EA'
0369 010E C14A      MOV  R10,ENTVAL  SET INITIAL ENTRY POINT
0370 0110 0207      LI   SPTR,ENDBUF START OF PXRMTN IS END USRMEM
      0112 00AC'
0371 0114 C487      MOV  SPTR,*ESTPT END OF SYMBOL TABLE POINTER
0372 0116 0607      DEC  SPTR
0373 0118 C4C7      MOV  SPTR,*SYMP  INIT SYMBOL TABLE POINTER
0374 011A 05A0      INC  @FSTIM      SET FIRST MODULE FLAG
      011C 041A'
0375 011E 0208      LI   TPPT,>80   SET DATA LIMIT
      0120 0080
0376          *
0377          * LIST OPTION
0378          *
0379 0122 020A      LI   IOPARM,LSTMSG PRINT LIST OPTION MESSAGE
      0124 007F'
0380 0126 0416      BLWP *PRNTN
0381 0128 C281      MOV  RDRBF,IOPARM WAIT FOR RESPONSE
0382 012A 0420      BLWP @KEYIN
      012C 062C'
0383 012E 04E0      CLR  @LIST      CLEAR LIST FLAG
      0130 053C'
0384 0132 9811      CB   *RDRBF,@F  IS FULL LIST DESIRED?
      0134 000A'
0385 0136 1602      JNE  ASK        NO - CONTINUE
0386 0138 05A0      INC  @LIST      YES - SET FULL LIST FLAG
      013A 053C'
0387          *
0388          * REQUEST LOAD/END ACTION
0389          *
0390 013C          ASK
0391 013C 02E0      LWPI MAINWP     LOAD WORKSPACE POINTER
      013E 009E'
0392 0140 06A0      BL   @PRCRLF    PRINT CR/LF
      0142 00DE'
0393 0144 1000      NOP            IGNORE ERROR RETURN
0394 0146          REQLE

```

0395	0146	020A	LI	IOPARM, LODMSG	PRINT LOAD/END MSG
	0148	0045'			
0396	014A	0416	BLWP	*PRNTN	
0397	014C	C281	MOV	RDRBF, IOPARM	
0398	014E	04DA	CLR	*IOPARM	INIT INPUT BUFFER TO 0
0399	0150	0420	BLWP	@KEYIN	WAIT FOR RESPONSE
	0152	062C'			
0400	0154	9811	CB	*RDRBF, @END	END?
	0156	000C'			
0401	0158	1314	JEQ	ENDPRO	YES
0402	015A	9811	CB	*RDRBF, @LOAD	LOAD?
	015C	000B'			
0403	015E	16F3	JNE	REQLE	NO, ASK AGAIN
0404	0160	C820	MOV	@K7, @CTLUN0	DEFAULT TO CASSETTE LUN0 7
	0162	0594'			
	0164	0012'			
0405	0166	06D1	SWPB	*RDRBF	CASSETTE LUN0 INPUT?
0406	0168	D011	MOVB	*RDRBF, TEMP	
0407	016A	1304	JEQ	OPNCAS	NO, USE DEFAULT
0408	016C	0240	ANDI	TEMP, >F00	CLEAR ALL BITS EXCEPT LUN0
	016E	0F00			
0409	0170	D800	MOVB	TEMP, @CTLUN0+1	SAVE INPUT LUN0
	0172	0013'			
0410	0174		OPNCAS		
0411	0174	C2A0	MOV	@CTLUN0, IOPARM	OPEN CASSETTE
	0176	0012'			
0412	0178	0420	BLWP	@OPEN	
	017A	060C'			
0413	017C		BATCH		
0414	017C	0420	BLWP	@DSPTCH	PROCESS MODULE
	017E	020C'			
0415	0180	10FD	JMP	BATCH	PROCESS NEXT MODULE
0416			*		
0417			*	PROCESS UNDEFINED SYMBOLS	
0418			*		
0419	0182		ENDPRO		
0420	0182	0709	SET0	UNDPT	SET UNDEFINED PRINT FLAG
0421	0184	C4C7	MOV	SPTR, *SYMP	POINT TO SYMBOL TABLE
0422	0186		CKSYMB		
0423	0186	8493	C	*SYMP, *ESTPT	END OF TABLE?
0424	0188	121B	JLE	ENTB	YES - END PROCESSING
0425	018A	0420	BLWP	@GETSFL	NO - GET SYMBOL FLAGS
	018C	05A8'			
0426	018E	061C	DEC	*SFLAGS	UNDEFINED?
0427	0190	1114	JLT	NXSYM	NO - GET NEXT SYMBOL
0428	0192	C249	MOV	UNDPT, UNDPT	PRINT 'UNDEFINED' MESSAGE?
0429	0194	1305	JEQ	SPRNT	NO - SKIP
0430	0196	020A	LI	IOPARM, UNDEF	POINT TO MESSAGE
	0198	002F'			
0431	019A	0420	BLWP	@PRINT	
	019C	0668'			
0432	019E	04C9	CLR	UNDPT	CLEAR PRINT FLAG
0433	01A0		SPRNT		
0434	01A0	0420	BLWP	@GETSNM	POINT TO SYMBOL NAME
	01A2	05BC'			
0435	01A4	C2A0	MOV	@SYMBOL+SYMMN+SYMMN, IOPARM	
	01A6	058E'			
0436	01A8	0200	LI	TEMP, UDSN	POINT TO NAME IN MESSAGE
	01AA	003A'			
0437	01AC	CC3A	MOV	*IOPARM+, *TEMP+	MOVE NAME INTO MESSAGE

```
0438 01AE CC3A      MOV  *IOPARM+, *TEMP+
0439 01B0 C41A      MOV  *IOPARM, *TEMP
0440 01B2 020A      LI   IOPARM, UDS      POINT TO MESSAGE
      01B4 0036
0441 01B6 0420      BLWP @PRINT          PRINT IT
      01B8 0668
0442 01BA          NXSYM
0443 01BA 64E0      S    @TEN, *SYMP     POINT TO NEXT ENTRY
      01BC 059A
0444 01BE 10E3      JMP  CKSYMB          CHECK NEW SYMBOL
0445
0446                * ENTRY MESSAGE AND TERMINATE OPTION
0447                *
0448 01C0          ENTB
0449 01C0 C284      MOV  ENTPT, IOPARM   PRINT ENTRY POINT MSG
0450 01C2 0420      BLWP @BINHEX        CONVERT ENTRY LOCATION
      01C4 0550
0451 01C6 00A8      DATA ENTVAL        ENTRY VALUE
0452 01C8 006C      DATA LOCNUM        LOCATION IN MESSAGE
0453 01CA 0420      BLWP @PRINT
      01CC 0668
0454 01CE 020A      LI   IOPARM, ENDACT PRINT TERM/CONT MSG
      01D0 0072
0455 01D2 0416      BLWP *PRNTN
0456 01D4 C281      MOV  RDRBF, IOPARM  WAIT FOR RESPONSE
0457 01D6 0420      BLWP @KEYIN
      01D8 062C
0458 01DA 9811      CB   *RDRBF, @TERM  TERMINATE?
      01DC 000D
0459 01DE 16AE      JNE  ASK            NO - GET MORE MODULES
0460                *
0461                * END OF PXRLAL
0462                *
0463 01E0 C805      MOV  ENTVAL, @USRPC SET UP ENTRY POINT FOR PXRMTN
      01E2 0000
0464 01E4 0380      RTWP              RETURN
```

```

0467 *
0468 * DSPTCH WORKSPACE AREA
0469 *
0470 01E6 DSWP
0471 01E6 0000 DATA 0 R0 INPT
0472 01E8 0000 DATA 0 R1 TGFLD
0473 01EA 00A0 BIASWD DATA >A0 R2 BIAS
0474 01EC DSWP2
0475 01EC 0000 NWLOC DATA 0 R3 NEWLOC
0476 01EE 0000 DATA 0 R4 LOCPT
0477 01F0 0000 DATA 0 R5 SVPT
0478 01F2 0000 DATA 0 R6 HLD
0479 01F4 0000 DATA 0 R7 FLGVL
0480 01F6 0000 DATA 0 R8 PC
0481 01F8 0000 DATA 0 R9 RETURN STATUS ON READ
0482 01FA 0000 DATA 0 R10 IOPARM
0483 01FC 0000 DATA 0 R11 RETURN
0484 01FE 0000 LOADPT DATA 0 R12 LDPT
0485 0200 03F6 DATA INCHK R13 INCP
0486 0202 0000 DATA 0 R14 INSID
0487 0204 00AE DATA TOPDAT R15 TDAT
0488 0206 0000 DATA 0 R13 OLD WP
0489 0208 0000 DATA 0 R14 OLD PC
0490 020A 0000 DATA 0 R15 OLD ST
    
```



```

0492 * TITLE: DSPTCH
0493 * LOAD MODULE DISPATCHER
0494 * REVISION: 05/01/74
0495 * ORIGINAL
0496 * 03/15/76
0497 * MODIFIED TO RUN WITH PXRMTX
0498 * COMPUTER: 990, 990 ASSEMBLY
0499 * ABSTRACT: OBJECT RECORDS ARE READ AND PROCESSED.
0500 * EACH RECORD CONTAINS SEVERAL OBJECT ENTRIES.
0501 * AN OBJECT ENTRY CONSISTS OF A TAG FIELD AND
0502 * UP TO TWO ADDITIONAL FIELDS. THE TAG FIELD
0503 * OF EACH ENTRY IS USED TO BRANCH TO THE
0504 * APPROPRIATE PROCESSOR. A TAG FIELD OF 'F'
0505 * BLWP @DSPTCH
0506 * STATISTICS:
0507 * WORKSPACE = DSWP AND DSWP2 (UNSHARED)
0508 * ALL SIXTEEN REGISTERS ARE USED BY OVERLAPPING
0509 * DSWP AND DSWP2. ALL ENTRANCES ARE DEFINED
0510 * THROUGH EXWP2 TO RETAIN THE RETURN ENVIRONMENT
0511 * THE FIRST INSTRUCTION IS THEN A 'LWPI DSWP'
0512 * TO GAIN THREE ADDITIONAL REGISTERS FOR USE.
0513 * WILL CAUSE A NEW RECORD TO BE READ.
0514 * CALLING SEQUENCE:
0515 * A 'LWPI DSWP2' WILL THEN PRECEDE ANY RTWP
0516 * INSTRUCTION.
0517 *
0518 * THE TAG FIELDS REPRESENT THE FOLLOWING
0519 * ENTRY TYPES:
0520 * TAG MEANING
0521 * 0 IDT RECORD
0522 * 1 ABSOLUTE ENTRY ADDRESS
0523 * 2 RELOCATABLE ENTRY ADDRESS
0524 * 3 EXTERNAL REFERENCE IN RELOCATABLE CODE
0525 * 4 EXTERNAL REFERENCE IN ABSOLUTE CODE
0526 * 5 RELOCATABLE EXTERNAL DEFINITION
0527 * 6 ABSOLUTE EXTERNAL DEFINITION
0528 * 7 CHECKSUM
0529 * 8 IGNORE CHECKSUM
0530 * 9 ABSOLUTE LOAD ADDRESS
0531 * A RELOCATABLE LOAD ADDRESS
0532 * B ABSOLUTE DATUM
0533 * C RELOCATABLE DATUM
0534 * D LOAD BIAS
0535 * E ILLEGAL TAG FIELD
0536 * F END OF RECORD
0537 * G RELOCATABLE SYMBOL (IGNORED)
0538 * H ABSOLUTE SYMBOL (IGNORED)
0539 *
0540 * ROUTINES CALLED:
0541 * BINHEX, CONVRT, CUMCHK, ERROR, GETSVL, HEXBIN,
0542 * INCHK, LDR, PROCLF, PRINTN, READ, SRCSYM
0543 *
0544 0200 DSPTCH
0545 0200 01EC DATA DSWP2 WORKSPACE
0546 020E 0210 DATA DSBG START
0547 0210 DSBG
0548 0210 02E0 LWPI DSWP ADJUST WORKSPACE
0549 0212 01E6
0549 0214 04CE CLR INSID
0550 *

```

```

0551          *      READ AN OBJECT RECORD (F)
0552          *
0553 0216      TAGF
0554 0216 0200      LI      INPT,LDBUF      POINT TO READ BUFFER
           0218 00A0
0555 021H 0280      MOV     INPT,IOPARM
0556 021C 0420      BLWP   @READ      READ A RECORD
           021E 0622
0557 0220 0249      ANDI   R9,EOF+IOERR    EOF OR IOERR ENCOUNTERED
           0222 6000
0558 0224 168B      JNE    ASK      YES, RESTART
0559          *
0560          *      ADJUST BIAS AND BRANCH BY TAG CHARACTER
0561          *
0562 0226      CHLP
0563 0226 0242      ANDI   BIAS,>FFFE      SET BIAS TO WORD BOUNDARY
           0228 FFFE
0564 022A 04C7      CLR    FLGVL      CLEAR REF/DEF FLAG
0565 022C 04C6      CLR    HLD      CLEAR TEMP LOCN FOR OFFSET
0566 022E 0070      MOVB   *INPT+,TGFLD    GET TAG FIELD
0567 0230 0221      AI     TGFLD,->3000    ADJUST FOR ASCII
           0232 0000
0568 0234 1120      JLT    TAGE      NOT IN RANGE - INV LD CD
0569 0236 0971      SRL   TGFLD,7      ADJUST TAG FOR BRANCH
0570 0238 0281      CI     TGFLD,JMPTBE-JMPTB-1    IN RANGE?
           023A 0031
0571 023C 151C      JGT    TAGE      NO - INV LD CD
0572 023E 0061      MOV    @JMPTB(TGFLD),TGFLD    BRANCH ON TAG
           0240 0244
0573 0242 0451      B      *TGFLD
0574          *
0575          *      TAG CHARACTER JUMP TABLE
0576          *
0577 0244      JMPTB
0578 0244 0288      DATA  TAG0      IDT RECORD
0579 0246 020E      DATA  TAG1      ABSOLUTE ENTRY ADDRESS
0580 0248 020C      DATA  TAG2      RELOCATABLE ENTRY ADDRESS
0581 024A 02D8      DATA  TAG3      EXTERNAL REF - RELOC CODE
0582 024C 02D8      DATA  TAG4      EXTERNAL REF - ABS CODE
0583 024E 034E      DATA  TAG5      RELOCATABLE EXTERNAL DEF
0584 0250 0350      DATA  TAG6      ABSOLUTE EXTERNAL DEF
0585 0252 03B8      DATA  TAG7      CHECKSUM
0586 0254 03BC      DATA  TAG8      IGNORE CHECKSUM
0587 0256 03C4      DATA  TAG9      ABSOLUTE LOAD ADDRESS
0588 0258 02BE      DATA  BKOUT    COLON: END OF MODUL
0589 025A 0276      DATA  TAGE      INVALID LOAD CODE
0590 025C 0276      DATA  TAGE      INVALID LOAD CODE
0591 025E 0276      DATA  TAGE      INVALID LOAD CODE
0592 0260 0276      DATA  TAGE      INVALID LOAD CODE
0593 0262 0276      DATA  TAGE      INVALID LOAD CODE
0594 0264 0276      DATA  TAGE      INVALID LOAD CODE
0595 0266 0302      DATA  TAGA      RELOCATABLE LOAD ADDRESS
0596 0268 03D2      DATA  TAGB      ABSOLUTE DATUM
0597 026A 03D0      DATA  TAGC      RELOCATABLE DATUM
0598 026C 03EA      DATA  TAGD      LOAD BIAS
0599 026E 0276      DATA  TAGE      ERROR
0600 0270 0216      DATA  TAGF      END OF RECORD
0601 0272 0282      DATA  TAGGH    RELOCATABLE SYMBOL
0602 0274 0282      DATA  TAGGH    ABSOLUTE SYMBOL
0603          0276      JMPTBE EQU $
    
```

```

0604      *
0605      *      ILLEGAL TAG CHARACTER (E)
0606      *
0607 0276      TAGE
0608 0276 020A      LI      IOPARM, ILCD      ILLEGAL LOAD CODE
           0278 2302
0609 027A      ERR
0610 027A 06A0      BL      @ERROR      CALL MONITOR ERROR ROUTINE
           027C 0000
0611 027E 0460      B      @LAL      FATAL ERROR - RESTART
           0280 00C8'

0612      *
0613      *      SYMBOL TABLE TAG CHARACTERS (G/H)
0614      *
0615 0282      TAGGH
0616 0282 0220      RI      INPT, 10      INCREMENT POINTER PAST ENTRY
           0284 000A
0617 0286 10CF      JMP      CHLP      TO IGNORE SYMBOL TABLE TAGS
0618      *
0619      *      PROCESS MODULE IDT (0)
0620      *
0621 0288      TAG0
0622 0288 C802      MOV      BIAS, @LOCPTR      GET PRESENT BIAS
           028H 0424'
0623 028C C38E      MOV      INSID, INSID      PRESENTLY INSIDE MODULE?
0624 028E 1607      JNE      TAG0B      YES - TEST IF PROPER
0625 0290 0420      BLWP    @LDR      NO - PROCESS IDT
           0292 0438'
0626 0294 1018      JMP      DSPEXT      EXIT IF MODULE FLUSHED
0627 0296 0460      B      @ASK      IF ERROR, EXIT LOOP
           0298 013C'
0628 029A 058E      INC      INSID      SET INSIDE MODULE FLAG
0629 029C 100D      JMP      TAG0CN      CONTINUE
0630 029E      TAG0B
0631 029E 06A0      BL      @CONVRT      GET CONVERTED VALUE
           02A0 0402'
0632 02A2 01EC'     DATA  NWLOC      STORE IT IN LOCATION TEMP
0633 02A4 A803      A      NEWLOC, @NWBIA
           02A6 04E2'
0634 02A8 9810      CB      *INPT, @BLANK      IS IDT NAME BLANK
           02AA 0008'
0635 02AC 1303      JEQ     TAG0C      YES - CONTINUE
0636      *
0637      *      MISSING END STATEMENT
0638      *
0639 02AE 020A      LI      IOPARM, MSEND      MISSING END STMT ERROR
           02B0 2303
0640 02B2 10E3      JMP      ERR
0641 02B4      TAG0C
0642 02B4 0220      RI      INPT, -4      ADJUST POINTER PAST LENGTH
           02B6 FFFC
0643 02B8      TAG0CN
0644 02B8 0220      RI      INPT, 12      ADJUST POINTER
           02BA 000C
0645 02BC 10B4      JMP      CHLP      GET NEXT TAG
0646      *
0647      *      END MODULE (:)
0648      *
0649 02BE      BKOUT
0650 02BE 04CE      CLR     INSID      CLEAR INSIDE MODULE FLAG

```

```

0651 02C0 A0A0      A   @NMBIAS,BIAS      ADD IN NEW BIAS
      02C2 04E2'
0652 02C4 0582      INC   BIAS              ACCOUNT FOR ODD LENGTH
0653 02C6          DSPEXT
0654 02C6 02E0      LWPI DSWP2
      02C8 01EC'
0655 02CA 0380      RTWP                    RETURN
0656
0657 *
0658 *   GET RELOCATABLE ENTRY ADDRESS (2)
0659 *
0659 02CC          TAG2
0660 02CC C182      MOV   BIAS,HLD          SET RELOCATABLE OFFSET
0661 *
0662 *   GET ABSOLUTE ENTRY ADDRESS (1)
0663 *
0664 02CE          TAG1
0665 02CE 0690      BL   *INCP              TEST INSIDE MODUL AND CONVERT
0666 02D0 00A8'     DATA ENTADD           PLACE IN ENTRY ADDRESS
0667 02D2 A806      A    HLD,@ENTADD       ADD IN OFFSET
      02D4 00A8'
0668 02D6 10A7     JMP   CHLP              GET NEXT TAG
0669 *
0670 *   GET RELOCATABLE CODE REF (3)
0671 *
0672 02D8          TAG3
0673 02D8 C182      MOV   BIAS,HLD          SET RELOCATABLE OFFSET
0674 *
0675 *   GET ABSOLUTE CODE REF (4)
0676 *
0677 02DA          TAG4
0678 02DA 0690      BL   *INCP              TEST INSIDE MODUL AND CONVERT
0679 02DC 01EC'     DATA NEWLOC           PLACE IN NEWLOC
0680 02DE A0C6      A    HLD,NEWLOC        ADJUST WITH BIAS
0681 *
0682 *   PROCESS REF CHAIN LINKAGE
0683 *
0684 02E0 C800      MOV   INPT,@NSMBL       POINT TO SYMBOL NAME
      02E2 04EA'
0685 02E4 0420      BLWP @SRCSYM           SEARCH SYMBOL TABLE
      02E6 05C6'
0686 02E8 101B     JMP   ENTERR           IF NOT FOUND, ENTER IT
0687 02EA 0220      AI   INPT,6            ADJUST POINTER
      02EC 0006
0688 02EE C103     MOV   NEWLOC,LOCPT      POINT TO NEW END OF CHAIN
0689 02F0 139A     JEQ  CHLP              IF ZERO, NO REF, SO END PROC
0690 02F2 0420      BLWP @GETSVL
      02F4 05B0'
0691 02F6 C160     MOV   @SYMBOL+SYMMN+SYMMN,SVPT GET VALUE
      02F8 058E'
0692 02FA 0620     DEC  @SYMBOL+FLAGS+FLAGS
      02FC 058C'
0693 02FE 1109     JLT  FXRF              IF DEF, SATISFY CHAIN
0694 0300          LPFXR
0695 0300 A10C      A    LDPT,LOCPT         POINT TO ACTUAL MEM ADDR
0696 0302 C194     MOV   *LOCPT,HLD        GET NEXT CHAIN ENTRY
0697 0304 1302     JEQ  ENDLPR            IF ZERO, END OF CHAIN
0698 0306 C106     MOV   HLD,LOCPT         CHANGE LOCATION POINTER
0699 0308 10FB     JMP  LPFXR             CONTINUE CHAINING
0700 030H          ENDLPR
0701 030A A10C      A    LDPT,LOCPT         POINT TO ACTUAL MEM ADDR

```

0702	030C	C515	MOV	*SVPT,*LOCPT	BREAK END LINK
0703	030E	C543	MOV	NEWLOC,*SVPT	CREATE NEW LINK IN SYMBOL TAB
0704	0310	108A	JMP	CHLP	END PROC, GET NEXT TAG
0705	0312		FXRF		
0706	0312	C0D5	MOV	*SVPT,NEWLOC	GET VALUE
0707	0314		LPDF		
0708	0314	A10C	A	LDPT,LOCPT	POINT TO ACTUAL MEM ADDR
0709	0316	C194	MOV	*LOCPT,HLD	POINT TO NEXT ENTRY IN CHAIN
0710	0318	C503	MOV	NEWLOC,*LOCPT	FILL IN RESOLVED VALUE
0711	031A	C106	MOV	HLD,LOCPT	CHANGE LOCATION POINTER
0712	031C	16FB	JNE	LPDF	CONTINUE CHAINING
0713	031E		CHLP3		
0714	031E	1083	JMP	CHLP	END PROC, GET NEXT TAG
0715			*		
0716			*	ENTER SYMBOL INTO TABLE	
0717			*		
0718	0320		ENTERR		
0719	0320	0587	INC	FLGVL	SET REF CODE
0720	0322		ENTERD		
0721	0322	C160	MOV	@SYMBOL+SYMPTR+SYMPTR,SVPT	
	0324	0588			
0722	0326	C547	MOV	FLGVL,*SVPT	MARK FLAG
0723	0328	0225	AI	SVPT,-7	POINT TO NAME LOCATION
	032A	FFF9			
0724	032C	DD70	MOVB	*INPT+,*SVPT+	
0725	032E	DD70	MOVB	*INPT+,*SVPT+	MOVE IN NAME
0726	0330	DD70	MOVB	*INPT+,*SVPT+	
0727	0332	DD70	MOVB	*INPT+,*SVPT+	
0728	0334	DD70	MOVB	*INPT+,*SVPT+	
0729	0336	D570	MOVB	*INPT+,*SVPT	
0730	0338	0225	AI	SVPT,-7	POINT TO VALUE
	033A	FFF9			
0731	033C	C543	MOV	NEWLOC,*SVPT	MOVE IN VALUE
0732	033E	C805	MOV	SVPT,@ENDST	CREATE NEW END OF SYMBOL TABLE
	0340	059C			
0733	0342	881F	C	*TDAT,@ENDST	IS LIMIT BELOW SYMBOL TABLE?
	0344	059C			
0734	0346	1AEB	JL	CHLP3	YES - GET NEXT TAG
0735			*		
0736			*	LOAD ADDRESS ERROR	
0737			*		
0738	0348		LDADER		
0739	0348	020A	LI	IOPARM,LAER	POINT TO ERROR CODE
	034A	2304			
0740	034C	1096	JMP	ERR	
0741			*		
0742			*	GET RELOCATABLE DEF (5)	
0743			*		
0744	034E		TAG5		
0745	034E	C182	MOV	BIAS,HLD	SET RELOCATABLE OFFSET
0746			*		
0747			*	GET ABSOLUTE DEF (6)	
0748			*		
0749	0350		TAG6		
0750	0350	069D	BL	*INCP	TEST INSIDE MODUL AND CONVERT
0751	0352	01EC	DATA	NWLOC	PLACE IN NWLOC
0752	0354	A0C6	A	HLD,NEWLOC	ADJUST WITH BIAS
0753			*		
0754			*	PROCESS DEF RESOLUTION	
0755			*		

0756	0356	0060	MOV	@LIST, TGFLD	IS FULL LIST DESIRED
	0358	053C			
0757	035H	1313	JEQ	PRNTOV	NO - SKIP
0758	035C	0420	BLWP	@BINHEX	YES - CONVERT LOCATION
	035E	0550			
0759	0360	01EC	DATA	NWLOC	LOCATION VALUE
0760	0362	002H	DATA	DEFL	LOCATION IN MESSAGE
0761	0364	0201	LI	TGFLD, DEFN	POINT TO NAME IN MESSAGE
	0366	0022			
0762	0368		MVNM		
0763	0368	DC70	MOV	*INPT+, *TGFLD+	MOVE IN NAME
0764	036A	0281	CI	TGFLD, DEFN+6	
	036C	0028			
0765	036E	11FC	JLT	MVNM	
0766	0370	0220	RI	INPT, -6	RESET POINTER
	0372	FFFA			
0767	0374	06A0	BL	@PRCRLF	PRINT CR/LF
	0376	0142			
0768	0378	1000	NOP		IGNORE ERROR RETURN
0769	037A	020A	LI	IOPARM, DEFM	PRINT DEFINITION MESSAGE
	037C	0014			
0770	037E	0420	BLWP	@PRINTN	
	0380	066C			
0771	0382		PRNTOV		
0772	0382	C800	MOV	INPT, @NSMBL	POINT TO SYMBOL NAME
	0384	04EA			
0773	0386	0420	BLWP	@SRCSYM	SEARCH SYMBOL TABLE
	0388	05C6			
0774	038A	10CB	JMP	ENTERD	NOT FOUND, ENTER IT
0775	038C	0220	RI	INPT, 6	ADJUST POINTER
	038E	0006			
0776	0390	0620	DEC	@SYMBOL+FLAGS+FLAGS	DEFINED?
	0392	058C			
0777	0394	1305	JEQ	SDF	NO - SET DEFINITION VALUE
0778			*		
0779			*	MULTIPLY DEFINED SYMBOL	
0780			*		
0781	0396	020A	LI	IOPARM, MULDF	
	0398	0042			
0782	039H	0420	BLWP	@PRINTN	
	039C	066C			
0783	039E		CHLP2		
0784	039E	10BF	JMP	CHLP3	CONTINUE
0785	03A0		SDF		
0786	03A0	0420	BLWP	@GETSVL	GET SYMBOL VALUE
	03A2	05B0			
0787	03A4	C160	MOV	@SYMBOL+SYMMN+SYMMN, SVPT	POINT TO VALUE
	03A6	058E			
0788	03A8	C115	MOV	*SVPT, LOCPT	POINT TO REF CHAIN
0789	03AA	C543	MOV	NEWLOC, *SVPT	CHANGE VALUE
0790	03AC	C160	MOV	@SYMBOL+SYMPTR+SYMPTR, SVPT	POINT TO ENTRY
	03AE	0588			
0791	03B0	04D5	CLR	*SVPT	CHANGE FLAG
0792	03B2	C104	MOV	LOCPT, LOCPT	WAS ACTUAL REF USED?
0793	03B4	13F4	JEQ	CHLP2	NO - FINISH PROCESSING
0794	03B6	10AE	JMP	LPDF	PROCESS CHAIN SATISFACTION
0795			*		
0796			*	GET CHECKSUM (?)	
0797			*		
0798	03B8		TAG7		

```

0799 03B8 0420          BLWP @CUMCHK          CHECK CUMULATIVE SUM
      03BA 04F2'
0800 *
0801 * SKIP CHECKSUM (8)
0802 *
0803 03BC TAG8
0804 03BC 0220          AI INPT,4          SKIP CUMSUM
      03BE 0004
0805 03C0 10EE          JMP CHLP2          GET NEXT FIELD
0806 *
0807 * GET RELOCATABLE LOAD ADDRESS (A)
0808 *
0809 03C2 TAGA
0810 03C2 C182          MOV BIAS,HL D      SET RELOCATABLE OFFSET
0811 *
0812 * GET ABSOLUTE LOAD ADDRESS (9)
0813 *
0814 03C4 TAG9
0815 03C4 069D          BL *INCP          TEST INSIDE MODUL AND CONVERT
0816 03C6 01EC'        DATA NWLOC        PLACE IN NWLOC
0817 03C8 00C6          R HLD,NEWLOC      ADJUST WITH BIAS
0818 03CA C203          MOV NEWLOC,PC     SET LOCATION COUNTER
0819 03CC A20C          R LDPT,PC         ADJUST WITH ACTUAL LOAD POINT
0820 03CE 10E7          JMP CHLP2        CONTINUE
0821 *
0822 * GET RELOCATABLE DATUM (C)
0823 *
0824 03D0 TAGC
0825 03D0 C182          MOV BIAS,HL D      SET RELOCATABLE OFFSET
0826 *
0827 * GET ABSOLUTE DATUM (B)
0828 *
0829 03D2 TAGB
0830 03D2 8808          C PC,@ENDST      IS ADDR PAST SYMBOL TABLE?
      03D4 059C'
0831 03D6 14B8          JHE LOADER        YES - ERROR
0832 03D8 821F          C *TDAT,PC       IS OLD LIMIT ABOVE ADDRESS?
0833 03DA 1401          JHE SETLP        YES - CONTINUE
0834 03DC C7C8          MOV PC,*TDAT     CREATE NEW LIMIT
0835 03DE SETLP
0836 03DE C808          MOV PC,@LOADV2   SET UP LOAD POINT
      03E0 03E4'
0837 03E2 069D          BL *INCP          TEST INSIDE MODUL AND CONVERT
0838 03E4 03E4'        LOADV2 DATA $    PLACE IN PC
0839 03E6 AE06          R HLD,*PC+       ADD IN OFFSET
0840 03E8 10DA          JMP CHLP2        GET NEXT TAG
0841 *
0842 * GET LOAD BIAS (D)
0843 *
0844 03EA TAGD
0845 03EA C38E          MOV INSID,INSID  INSIDE MODUL?
0846 03EC 1606          JNE INER         YES - ILLEGAL LD SEQUENCE
0847 03EE 06A0          BL @CONVRT       GET CONVERTED VALUE
      03F0 0402'
0848 03F2 01EA'        DATA BIASWD     PLACE IN BIAS
0849 03F4 10D4          JMP CHLP2        GET NEXT TAG
0850 *
0851 * TEST IF INSIDE MODUL PROCESSING
0852 *
0853 03F6 INCHK
  
```

```

0804 03F6 C38E      MOV  INSID,INSID      INSIDE MODUL?
0805 03F8 1604      JNE  CONVRT          YES - CONVERT
0806                *
0807                *      OUTSIDE MODULE
0808                *
0809 03FH          INER
0860 03FA 020A      LI   IOPARM,ILSQ     ILLEGAL LOAD SEQUENCE ERROR
      03FC 2301
0861 03FE 0460      B    @ERR
      0400 027A
0862                *
0863                *      CONVERT FIELD IN OBJECT CODE
0864                *
0865 0402          CONVRT
0866 0402 C83B      MOV  *11+,@CONLOC    MOVE IN CONVERSION LOCATION
      0404 0410
0867 0406 C800      MOV  INPT,@CONVL     POINT TO BUFFER AREA
      0408 040E
0868 040A 0420      BLWP @HEXBIN        CONVERT TO BINARY
      040C 056C
0869 040E 040E     CONVL  DATA $       DATA FOR CONVERSION
0870 0410 0410     CONLOC DATA $     PLACE IN PROPER LOCATION
0871 0412 0220      RI   INPT,4        ADJUST POINTER
      0414 0004
0872 0416 045B      RT                    RETURN

```



```
*  
* LDR WORKSPACE AREA  
*  
0875 0418 LDWP  
0876 0418 0218' DATA LDBUF R0 RD  
0877 0418 0000 FSTIM DATA 0 R1 FST  
0878 041C 0000 DATA 0 R2 PTR  
0879 041E 04EA' DATA NSMBL R3 NSMBLP  
0880 0420 0000 DATA 0 R4 SYMB  
0881 0422 0000 DATA 0 R5 TMP  
0882 0424 0000 LOCPTR DATA 0 R6 LOC  
0883 0426 0506' DATA SRCSYM R7 SRSM  
0884 0428 0050' DATA NAMMSG R8 NMMSG  
0885 042A 0000 DATA 0 R9  
0886 042C 0000 DATA 0 R10 IOPARM  
0887 042E 0000 DATA 0 R11 RTN  
0888 0430 056C' DATA HEXBIN R12 HEXP2  
0889 0432 0000 DATA 0 R13 OLD WP  
0890 0434 0000 DATA 0 R14 OLD PC  
0891 0436 0000 DATA 0 R15 OLD ST
```

```

0896 * TITLE: LDR
0897 * LOADER PROCESS OF IDT ENTRY
0900 * REVISION: 05/01/74
0901 * ORIGINAL
0902 * 03/15/76
0903 * MODIFIED TO RUN WITH PXRMTX
0904 * COMPUTER: 990, ASSEMBLY
0905 * ABSTRACT: AN IDT ENTRY IS CHECKED. IF THIS IS
0906 * THE FIRST MODULE, THE NAME WILL BE PLACED
0907 * IN THE SYMBOL TABLE AND CONTROL RETURNED
0908 * TO THE CALLING ROUTINE FOR MODULE PROCESSING.
0909 * IF THIS IS A LATER ENTRY WHOSE NAME IS NOT
0910 * IN THE SYMBOL TABLE, IT WILL BE FLUSHED.
0911 * IF ALREADY DEFINED, AN ERROR MESSAGE WILL
0912 * BE PRINTED AND THE MODULE FLUSHED. IF ONLY
0913 * REF'D, MODULE WILL BE PROCESSED AS FIRST
0914 * MODULE, AND THE NAME WILL BE MARKED AS DEF'D.
0915 * NO VALUE IS GIVEN TO THE NAME SO IT MAY NOT
0916 * BE USED TO SATISFY REFERENCES. THREE RETURNS
0917 * ARE PROVIDED FOR ACCEPTABLE MODULES, MODULES
0918 * FLUSHED BECAUSE OF ERROR, AND MODULES FLUSHED
0919 * BECAUSE NOT REF'D.
0920 * CALLING SEQUENCE:
0921 * BLWP @LDR
0922 * NOT REF'D MODULE FLUSHED - 1 WD INSTR
0923 * ERROR - MODULE FLUSHED - 2 WD INSTR
0924 * ACCEPTABLE MODULE - 1 WD INSTR
0925 * STATISTICS: WORKSPACE = LDWP (UNSHARED)
0926 * ROUTINES CALLED:
0927 * BINHEX, ERROR, GETSVL, HEXBIN, PROCRLF, PRINTN,
0928 * READ, SRCSYM
0929 0438 LDR
0930 0438 0418 DATA LDWP WORKSPACE
0931 043H 043C DATA LDBG START
0932 043C LDBG
0933 043C 0080 MOV RD, PTR POINT TO READ BUFFER
0934 043E 0582 INC PTR SKIP TAG
0935 *
0936 * CONVERT RELOCATABLE LENGTH
0937 *
0938 0440 0802 MOV PTR, @PTRPT POINT TO BUFFER
0939 0442 0446 PTRPT
0940 0444 041C BLWP *HEXP2 CONVERT LENGTH
0941 0446 0446 DATA $ POINTER TO MODUL LENGTH
0942 0448 04E2 DATA NMBIAS CREATE NEW BIAS
0943 044A 0222 AI PTR, 4 ADJUST PAST LENGTH
0944 044C 0004
0945 *
0946 * TEST FOR LOADABLE MODUL
0947 *
0948 044E 0041 MOV FST, FST FIRST TIME?
0949 0450 1616 JNE FIRST YES - PROCESS
0950 0452 04C2 MOV PTR, *NSMBLP
0951 0454 0417 BLWP *SRSM NO, SEARCH FOR NAME
0952 0456 1008 JMP FLCL SYMBOL NOT FOUND - FLUSH
0953 0458 0620 DEC @SYMBOL+FLAGS+FLAGS PREVIOUSLY DEFINED?
0954 045H 058C
0955 045C 131D JEQ MRKDF NO, MARK AS DEF
0956 *
0957 * PREVIOUSLY LOADED MODULE ERROR

```

```

0953          *
0954 045E 020A          LI  IOPARM, PRLM          PREV LOAD MODULE ERROR
          0460 2305
0955 0462 06A0          BL  @ERROR          CALL MONITOR ERROR PROCESSOR
          0464 027C
0956 0466 05CE          INCT R14          TAKE ERROR EXIT
0957          *
0958          *          FLUSH MODULE
0959          *
0960 0468          FLCL
0961 0468 C080          MOV  RD, PTR          POINT TO BUFFER AREA
0962 046A          LPFL
0963 046A C280          MOV  RD, IOPARM
0964 046C 0420          BLWP @READ          READ A RECORD
          046E 0622
0965 0470 0249          ANDI R9, EOF+IOERR          EOF OR I/O ERROR?
          0472 0000
0966 0474 1603          JNE  FLRT
0967 0476 9812          CB   *PTR, @COLON          FIRST CHAR = :
          0478 0009
0968 047A 16F7          JNE  LPFL          NO- CONTINUE
0969 047C          FLRT
0970 047C 0380          RTWP          RETURN
0971          *
0972          *          PROCESS FIRST MODUL
0973          *
0974 047E          FIRST
0975 047E 04C1          CLR  FST          CLEAR FIRST MODUL FLAG
0976 0480 0205          LI  TMP, 6          INIT COUNTER
          0482 0006
0977 0484 6820          S   @TEN, @ENDST          SET NEW END OF SYMBOL TABLE
          0486 059A
          0488 059C
0978 048A C120          MOV  @SYMTAB, SYMB          POINT TO SYMBOL NAME LOCATION
          048C 00AC
0979 048E 0224          AI  SYMB, -7
          0490 FFF9
0980 0492          LP
0981 0492 DD32          MOVB *PTR+, *SYMB+          MOVE IN MODUL NAME
0982 0494 0605          DEC  TMP
0983 0496 15FD          JGT  LP
0984          *
0985          *          PROCESS ACCEPTABLE MODUL
0986          *
0987 0498          MRKDF
0988 0498 05C2          INCT PTR          SKIP LAST TWO CHARACTERS
0989 049A C120          MOV  @SYMBOL+SYMPTR+SYMPTR, SYMB
          049C 0588
0990 049E 04D4          CLR  *SYMB          MARK AS DEF
0991 04A0 0224          AI  SYMB, -7          POINT TO NAME
          04A2 FFF3
0992 04A4 0205          LI  TMP, NAME          MOVE NAME TO OUTPUT LINE
          04A6 0056
0993 04A8 CD74          MOV  *SYMB+, *TMP+
0994 04AA CD74          MOV  *SYMB+, *TMP+
0995 04AC CD74          MOV  *SYMB+, *TMP+
0996 04AE 0420          BLWP @GETSVL          POINT TO SYMBOL VALUE
          04B0 05B0
0997 04B2 C120          MOV  @SYMBOL+SYMMN+SYMMN, SYMB
          04B4 058E
    
```

0998 04B6 0506	MOV LOC,*SYMB	MOVE IN LOC VALUE
0999 04B8 0420	BLWP @BINHEX	CONVERT LOC FOR PRINT
04B8 0550		
1000 04BC 0424	DATA LOCPTR	LOCATION VALUE
1001 04BE 000E	DATA LOCVAL	PRINTING LOCATION
1002 04C0 06A0	BL @PRCRLF	PRINT CR/LF
04C2 0376		
1003 04C4 1000	NOP	IGNORE ERROR RETURN
1004 04C6 0288	MOV NMMSG,IOPARM	PRINT MODULE NAME MESSAGE
1005 04C8 0420	BLWP @PRINTN	
04CA 066C		
1006 04CC 022E	AI R14,6	TAKE ALTERNATE RETURN
04CE 0006		
1007 04D0 0380	RTWP	RETURN

```
1010 *
1011 * CHECKSUM CHECK WORKSPACE AREA
1012 *
1013 0402 CMWP
1014 0402 0500' DATA CKLOC R0 CKL
1015 0404 0000 DATA 0 R1 HVAL
1016 0406 0418' DATA LDBUF R2 RDRL
1017 0408 0000 DATA 0 R3 CUMPT
1018 040A 01E6' DATA DSWP+INPT+INPT R4 ENDOB
1019 040C 056C' DATA HEXBIN R5 HEXP
1020 040E 0000 RLVL DATA 0 R6 RLRG
1021 04E0 0000 DATA 0 R7 CSUM
1022 04E2 0000 NWBIAS DATA 0 R8
1023 04E4 0000 DATA 0 R9
1024 04E6 0000 DATA 0 R10 IOPARM
1025 04E8 0000 DATA 0 R11 RETURN
1026 04EA 0000 NSMBL DATA 0 R12
1027 04EC 0000 DATA 0 R13 OLD WP
1028 04EE 0000 DATA 0 R14 OLD PC
1029 04F0 0000 DATA 0 R15 OLD ST
```

```

1031      * TITLE:      CUMCHK
1032      *              CHECK CHECKSUM VALUE
1033      * REVISION:    05/01/74
1034      *              ORIGINAL
1035      *              03/15/76
1036      *              MODIFIED TO RUN WITH PXRMT
1037      * COMPUTER:    990, ASSEMBLY
1038      * ABSTRACT:    THE CHECKSUM IS COMPUTED AND COMPARED
1039      *              TO THE VALUE PRESENT ON THE OBJECT
1040      *              RECORD. IF AN ERROR OCCURS, A MESSAGE
1041      *              IS PRINTED AND KEYBOARD RESPONSE IS
1042      *              REQUIRED BEFORE THE NEXT INPUT RECORD
1043      *              MAY BE READ.
1044      * CALLING SEQUENCE:
1045      *              BLWP @CUMCHK
1046      * STATISTICS:   WORKSPACE = CMWP (UNSHARED)
1047      * ROUTINES CALLED:
1048      *              ERROR, HEXBIN, KEYIN, OPEN
1049 04F2    CUMCHK
1050 04F2 04D2' DATA CMWP      WORKSPACE
1051 04F4 04F6' DATA CMBG      START
1052      *
1053      * COMPUTE CHECKSUM
1054      *
1055 04F6    CMBG
1056 04F6 04C7 CLR CSUM      CLEAR CHECKSUM
1057 04F8 C0C2 MOV RDRL, CUMPT    POINT TO START OF RECORD
1058 04FA    CKCOM
1059 04FA 04C1 CLR HVAL      CLEAR VALUE
1060 04FC D073 MOVB *CUMPT+, HVAL  GET ASCII CHARACTER
1061 04FE 06C1 SWPB HVAL      RIGHT JUSTIFY
1062 0500 A1C1 A HVAL, CSUM  ADD TO CHECKSUM
1063 0502 8503 C CUMPT, *ENDOB    FINISHED RECORD AREA?
1064 0504 11FA JLT CKCOM      NO - CONTINUE
1065 0506 0507 NEG CSUM      YES - NEGATE VALUE FOR CHECKSUM
1066      *
1067      * COMPARE TO VALUE ON RECORD
1068      *
1069 0508 C403 MOV CUMPT, *CKL    POINT TO VALUE ON RECORD
1070 050A 0415 BLWP *HEXP      CONVERT TO BINARY VALUE
1071 050C 050C' CKLOC DATA $      CONVERSION LOCATION
1072 050E 04DE' DATA RLVL      POSITION FOR RESULT
1073 0510 81C6 C RLRG, CSUM    COMPARE THE TWO
1074 0512 130D JEQ OUTCK      EQUAL - EXIT
1075      *
1076      * BAD CHECKSUM PROCESSOR
1077      *
1078 0514 020A LI IOPARM, CKSM    BAD CHECKSUM ERROR
1079 0516 2306      0518 06A0 BL @ERROR      CALL MONITOR ERROR PROCESSOR
1080 051A 0464'      051C C282 MOV RDRL, IOPARM  WAIT FOR RESPONSE
1081 051E 0420 BLWP @KEYIN      FROM KEYBOARD
1082 0522 C2A0 MOV @CTLUNG, IOPARM  OPEN CASSETTE
1083 0524 0012'      0526 0420 BLWP @OPEN
1084 0528 060C'      052A 020E LI 14, TAGF
1085 052C 0216'

```

PXRLAL

SDSMAC 947075 *B 16:59:48 SATURDAY, DEC 11, 1976.

WILSONSUN CHECK

948926-9901 **

PAGE 0029

1080 05ZE

OUTCK

1000 05ZE 0380

RTWP

```
1089 *
1090 * CONVERSION ROUTINE WORKSPACE AREA
1091 *
1092 0530 CONVMS
1093 0530 0000 BINVL DATA 0 R0 BINVAL BINARY VALUE
1094 0532 0000 DATA 0 R1 VALSV TEMP VALUE SAVE
1095 0534 0000 DATA 0 R2 MADD LOCN HEX VALUE
1096 0536 0000 SVCBLK DATA 0 R3 SCCODE SUPV CALL CODE
1097 0538 0000 HSCVL DATA 0 R4 ASCII VALUE
1098 053A 0000 DATA 0 R5
1099 053C 0000 LIST DATA 0 R6
1100 053E 0C00 DATA >C00 R7 BHCODE BINARY/HEX SCC
1101 0540 0D00 DATA >D00 R8 HBCODE HEX/BINARY SCC
1102 0542 0000 DATA SVCALT R9 SVCAL1 PXRMTN SUPV CALL ROUT
1103 0544 0536 DATA SVCBLK R10 SUPV CALL BLOCK PTR
1104 0546 FFFC DATA -4 R11 HCNT
1105 0548 0000 DATA 0 R12 FCNT
1106 054A 0000 DATA 0 R13 OLD WP
1107 054C 0000 DATA 0 R14 CHOPC OLD PC
1108 054E 0000 DATA 0 R15 OLD ST
```



```

1110      * TITLE:      BINHEX/HEXBIN
1111      *              CONVERSION ROUTINES
1112      * REVISION:   05/01/74
1113      *              ORIGINAL
1114      *              03/15/76
1115      *              MODIFIED TO USE PXRMTN CONVERSION ROUTINES
1116      * COMPUTER:   990, ASSEMBLY
1117      * ABSTRACT:   SETS UP SUPERVISOR CALL BLOCK FOR CALL
1118      *              AND STORES VALUES AFTER RETURN IN USER
1119      *              DEFINED LOCATIONS.
1120      * CALLING SEQUENCE:
1121      *              BINARY TO HEX ASCII
1122      *              BLWP @BINHEX
1123      *              DATA ADDRESS-OF-VALUE
1124      *              DATA ADDRESS-FOR-PLACEMENT
1125      *
1126      *              HEX ASCII TO BINARY
1127      *              BLWP @HEXBIN
1128      *              DATA ADDRESS-OF-VALUE
1129      *              DATA ADDRESS-FOR-PLACEMENT
1130      * ROUTINES CALLED: SVCALT (EXT REF IN PXRMTN)
1131      *
1132 0550      BINHEX
1133 0550 0530'  DATA CONVMS      WORKSPACE
1134 0552 0554'  DATA BIHEX      START
1135 0554      BIHEX
1136 0554 C03E   MOV  *CHOPC+,BINVAL    POINT TO ADDRESS OF VALUE
1137 0556 C010   MOV  *BINVAL,BINVAL    GET VALUE IN R0 FOR CALL
1138 0558 C0C7   MOV  HBCODE,SCCODE   SET UP SUPV CALL CODE
1139 055A 0419   BLWP *SVCAL1      BINARY TO HEX SUPV CALL
1140 055C C0BE   MOV  *CHOPC+,MADD    GET PLACEMENT LOC & NEW RTN
1141 055E 0201   LI   VALSV,ASCVL    SET UP ASCII VALUE POINTER
1142      0560 0538'
1142 0562 C30B   MOV  HCNT,PCNT      INIT COUNT TO 4
1143 0564 DCB1   MOVB *VALSV+,*MADD+  MOVE VALUE TO PLACEMENT LOCN
1144 0566 058C   INC  PCNT
1145 0568 11FD   JLT  BINH
1146 056H 0380   RTWP
1147
1148 056C      HEXBIN
1149 056C 0530'  DATA CONVMS      WORKSPACE
1150 056E 0570'  DATA HEXBI      START
1151 0570      HEXBI
1152 0570 C0BE   MOV  *CHOPC+,MADD    GET ADDRESS OF VALUE
1153 0572 C0C8   MOV  HBCODE,SCCODE   SET UP SUPERVISOR CALL CODE
1154 0574 0201   LI   VALSV,ASCVL    SET UP ASCII VALUE POINTER
1155      0576 0538'
1155 0578 C30B   MOV  HCNT,PCNT      INIT COUNT TO 4
1156 057A DC72   MOVB *MADD+,*VALSV+  MOV HEX ASCII VALUE TO SCBLK
1157 057C 058C   INC  PCNT      CONTINUE UNTIL ALL
1158 057E 11FD   JLT  HEXB      BYTES MOVED
1159 0580 0419   BLWP *SVCAL1      HEX TO BINARY SUPV CALL
1160 0582 C07E   MOV  *CHOPC+,VALSV    GET ADDR FOR PLACEMENT LOCN
1161 0584 C440   MOV  BINVAL,*VALSV    MOVE BINARY VALUE TO LOCN
1162 0586 0380   RTWP      RETURN

```

Address	Hex	Symbol	Value	Register	Symbol
1165	0588				
1166	0588	0000	DATA 0	WR0	SYMPTR
1167	058A	0000	DATA 0	WR1	SYMNO
1168	058C	0000	DATA 0	WR2	FLAGS
1169	058E	0000	DATA 0	WR3	SYMNM
1170	0590	0000	DATA 0	WR4	NSYM
1171	0592	0000	DATA 0	WR5	VALUE
1172	0594	0007	DATA 7	WR6	SEVEN
1173	0596	00AC	DATA SYMTAB	WR7	STPTR
1174	0598	0006	DATA 6	WR8	NWSCT
1175	059A	000A	DATA 10	WR9	TN
1176	059C	0000	DATA 0	WR10	NSBIT
1177	059E	0000	DATA 0	WR11	RETURN
1178	05A0	0009	DATA 9	WR12	NINE
1179	05A2	0000	DATA 0	WR13	OLD WP
1180	05A4	0000	DATA 0	WR14	OLD PC
1181	05A6	0000	DATA 0	WR15	OLD ST

```

1183 * TITLE: SENTRY
1184 * SYMBOL TABLE ENTRY VALUES
1185 * REVISION: 05/01/74
1186 * ORIGINAL
1187 * COMPUTER: 990, ASSEMBLY
1188 * ABSTRACT: THREE ROUTINES ARE USED TO PROVIDE:
1189 * 1. SYMBOL FLAG VALUE
1190 * 2. POINTER TO SYMBOL NAME
1191 * 3. POINTER TO VALUE AND VALUE
1192 * CALLING SEQUENCE:
1193 * BLWP @GETSFL
1194 * BLWP @GETSVL
1195 * BLWP @GETSUM
1196 * STATISTICS: WORKSPACE = SYMBOL (SHARED WITH SRCSYM)
1197 * SYMBOL TABLE ENTRY APPEARS AS FOLLOWS:
1198 * S Y
1199 * M B
1200 * O L
1201 * 0,1
1202 * WHERE 0 = DEFINED, 1 = UNDEFINED
1203 *
1204 * GET SYMBOL FLAG
1205 *
1206 05A8 GETSFL
1207 05A8 0588' DATA SYMBOL WORKSPACE
1208 05AH 05AC' DATA GTSFL START
1209 05AC GTSFL
1210 05AC C090 MOV *SYMPTR, FLAGS GET FLAGS WORD
1211 05AE 0380 RTWP RETURN
1212 *
1213 * GET SYMBOL VALUE
1214 *
1215 05B0 GETSVL
1216 05B0 0588' DATA SYMBOL WORKSPACE
1217 05B2 05B4' DATA GTSVL START
1218 05B4 GTSVL
1219 05B4 C0C0 MOV SYMPTR, SYMNM POINT TO START OF SYMBOL ENTRY
1220 05B6 60C0 S NINE, SYMNM POINT TO VALUE
1221 05B8 C153 MOV *SYMNM, VALUE GET SYMBOL VALUE
1222 05BA 0380 RTWP RETURN
1223 *
1224 * GET SYMBOL NAME
1225 *
1226 05BC GETSNM
1227 05BC 0588' DATA SYMBOL WORKSPACE
1228 05BE 05C0' DATA GTSNM START
1229 05C0 GTSNM
1230 05C0 C0C0 MOV SYMPTR, SYMNM POINT TO SYMBOL ENTRY
1231 05C2 60C6 S SEVEN, SYMNM POINT TO NAME
1232 05C4 0380 RTWP RETURN
    
```

```

1235 * TITLE: SRCSYM
1236 * SEARCH SYMBOL TABLE
1237 * REVISION: 05/01/74
1238 * ORIGINAL
1239 * 03/15/76
1240 * MODIFIED TO RUN WITH PXRMTX
1241 * COMPUTER: 990, ASSEMBLY
1242 * ABSTRACT: THE SYMBOL TABLE IS SEARCHED FOR A
1243 * GIVEN SYMBOL NAME. IF THE SYMBOL
1244 * IS FOUND, AN ALTERNATE EXIT IS
1245 * TAKEN. THE TABLE IS SEARCHED LINEARLY.
1246 * CALLING SEQUENCE:
1247 * BLWP @SRCSYM
1248 * SYMBOL-NOT-FOUND INSTRUCTION
1249 * SYMBOL-FOUND INSTRUCTION
1250 * STATISTICS: WORKSPACE = SYMBOL (SHARED WITH SENTRY)
1251 0506 SRCSYM
1252 0506 0588' DATA SYMBOL WORKSPACE
1253 0508 05CA' DATA SRC START
1254 *
1255 * POINT TO BEGINNING OF SYMBOL TABLE
1256 *
1257 050A SRC
1258 050A C017 MOV *STPTR, SYMPTR POINT TO SYMBOL TABLE
1259 *
1260 * COMPARE TO ENTRY
1261 *
1262 050C SYMCHK
1263 050C C120 MOV @NSMBL, NSYM POINT TO NEW SYMBOL
1264 050E 04EA' C SYMPTR, NSBIT PAST END OF TABLE?
1265 05D2 1209 JLE EXIT YES - EXIT
1266 05D4 C0C0 MOV SYMPTR, SYMNM POINT TO ENTRY
1267 05D6 60C6 S SEVEN, SYMNM POINT TO NAME
1268 05D8 C048 MOV NWSCT, SYMNO GET COUNT
1269 05DA COMPAR
1270 05DA 9D33 CB *SYMNM+, *NSYM+ ARE CHARACTERS THE SAME?
1271 05DC 1605 JNE NOTSAM NO - GET NEXT SYMBOL
1272 05DE 0601 DEC SYMNO ARE CHARACTERS FINISHED
1273 05E0 16FC JNE COMPAR NO, KEEP COMPARING
1274 05E2 C090 MOV *SYMPTR, FLAGS YES - GET FOUND SYMBOL'S FLAGS
1275 05E4 05CE INCT 14 TAKE ALTERNATE RETURN
1276 05E6 EXIT
1277 05E6 0380 RTWP RETURN
1278 *
1279 * LOOK AT NEXT SYMBOL
1280 *
1281 05E8 NOTSAM
1282 05E8 6009 S TN, SYMPTR POINT TO NEXT SYMBOL
1283 05EA 10F0 JMP SYMCHK GET NEXT SYMBOL
    
```

```
1286      *
1287      *      WORKSPACE AREA FOR INPUT/OUTPUT ROUTINES
1288      *
1289 05EC      IOWKS
1290 05EC 0900      DATA RDCOD      R0      RDCODP
1291 05EE 0B00      DATA WRCOD      R1      WRCODP
1292 05F0      SCBLK  BSS 12      R2-R7    I/O SUPV CALL BLK (PRB)
1293 05FC 0050      DATA 80      R8      LEN
1294 05FE 0542      DATA SVCALT      R9      SVCAL2
1295 0600 05F0      DATA SCBLK      R10     POINTER TO SUPV CALL BLOCK
1296 0602      BSS 10      R11-R15
```

1298 * TITLE: OPEN
1299 * ASSIGN A DEVICE TO A TASK
1300 * REVISION: 03/15/76
1301 * ORIGINAL
1302 * COMPUTER: 990, ASSEMBLY
1303 * ABSTRACT: SETS UP PRB AND MAKES SUPERVISOR
1304 * CALL TO OPEN A DEVICE
1305 * CALLING SEQUENCE:
1306 * R10 = LUNO
1307 * BLWP @OPEN
1308 * STATISTICS: WORKSPACE = IOWKS (SHARED WITH I/O ROUTINES)

1309 060C OPEN
1310 060C 05EC DATA IOWKS, \$+2 TRANSFER VECTOR
1310 060E 0610
1311 0610 04C3 CLR IOPLUN SET UP I/O OP
1312 0612 04C2 CLR IOC INIT CODE FOR I/O SUPV CALL
1313 0614 C30D MOV R13, R12 GET CALLER'S WORKSPACE PTR
1314 0616 022C AI R12, R10+R10 INDEX TO CALL PARAMETER
1314 0618 0014
1315 061A C31C MOV *R12, R12 GET LUNO
1316 061C E0CC SOC R12, IOPLUN OR LUNO INTO IOPLUN
1317 061E 0419 BLWP *SVCAL2 MAKE SUPV CALL
1318 0620 0380 RTWP RETURN

```
1320      * TITLE:      READ
1321      *              READ A RECORD
1322      * REVISION: 03/15/76
1323      *              ORIGINAL
1324      * COMPUTER: 990, ASSEMBLY
1325      * ABSTRACT: SETS UP PRB AND MAKES SUPV CALL
1326      *              TO READ A RECORD
1327      * CALLING SEQUENCE:
1328      *              R10 = BUFFER ADDRESS
1329      *              BLWP @READ
1330      * RETURN PARAMETERS:
1331      *              R10 = BUFFER ADDR+CHAR COUNT
1332      *              R9  = FLAGS
1333      * STATISTICS: WORKSPACE = IOWKS (SHARED WITH I/O ROUTINES)
1334 0622      READ
1335 0622 05EC'      DATA IOWKS, #+2          TRANSFER VECTOR
1336      0624 0626'
1336 0626 C0E0      MOV @CTLUN0, IOPLUN      STORE LUN0
1336      0628 0012'
1337 062A 1004      JMP KEYRD
```

```
1339      * TITLE:      KEYIN
1340      *              KEYBOARD INPUT ROUTINE
1341      * REVISION:   03/15/76
1342      *              ORIGINAL
1343      * COMPUTER:   990, ASSEMBLY
1344      * ABSTRACT:   SETS UP PRB AND MAKES SUPV CALL
1345      *              TO INPUT A RECORD FROM KEYBOARD
1346      * CALLING SEQUENCE:
1347      *              R10 = BUFFER ADDRESS
1348      *              BLWP @KEYIN
1349      * RETURN PARAMETERS:
1350      *              R10 = BUFFER ADDRESS+CHAR COUNT
1351      *              R9  = FLAGS
1352      * STATISTICS: WORKSPACE = IOWKS (SHARED WITH I/O ROUTINES)
1353 062C      KEYIN
1354 062C 05EC'   DATA IOWKS, $+2      TRANSFER VECTOR
1355 0630 C0E0   MOV  @KBLUN0, IOPLUN      STORE LUN0
1356 0634      KEYRD
1357 0634 D0C0   MOVB RDCODP, IOPLUN      STORE I/O OP
1358 0636 C30D   MOV  R13, R12              GET CALLER'S WORKSPACE PTR
1359 0638 022C   AI    R12, R10+R10      INDEX TO CALLING PARAMETERS
1360 063C C15C   MOV  *R12, BUFADR        SET UP BUFFER ADDRESS
1361 063E C188   MOV  LEN, BUFLN          SET UP BUFFER LENGTH
1362 0640 04C4   CLR  FLGS              INIT FLAGS
1363 0642 04C7   CLR  CHRCNT          INIT CHAR CNT TO ZERO
1364 0644 04C2   CLR  IOC              INIT CODE FOR I/O SUPV CODE
1365 0646 0419   BLWP *SVCAL2          MAKE SUPV CALL
1366 0648 A707   A    CHRCNT, *R12        INCR BUFF PTR TO END OF DATA
1367 064A 064C   DECT R12
1368 064C C704   MOV  FLGS, *R12        MOVE FLAGS TO CALLING WS
1369 064E 0243   ANDI IOPLUN, >00FF      IS THIS KEYBOARD READ?
1370 0652 8803   C    IOPLUN, @KBLUN0
1371 0654 000E'
1372 0656 1301   JEQ  KEYPLF            IF READ, RETURN
1373 0658 0380   RTWP
1374 065A      KEYPLF
1375 065A C0E0   MOV  @PRLUN0, IOPLUN      SETUP UP TO PRINT LF/CR
1376 065C 0010'
1377 065E D0C1   MOVB WRCODP, IOPLUN
1378 0660 0205   LI  BUFADR, LFCR
1379 0662 0006'
1377 0664 04C4   CLR  FLGS
1378 0666 100E   JMP  PRTKEY
```



```

1380 * TITLE: PRINT/PRINTN
1381 * PRINT A RECORD
1382 * REVISION: 03/15/76
1383 * ORIGINAL
1384 * COMPUTER: 990, ASSEMBLY
1385 * ABSTRACT: SETS UP PRB AND MAKES SUPV CALL
1386 * TO PRINT A RECORD
1387 * PRINT - PRINTS RECORD WITH CR
1388 * PRINTN- PRINTS RECORD WITHOUT CR
1389 * CALLING SEQUENCE:
1390 * R10 = BUFFER ADDRESS
1391 * BLWP @PRINT
1392 * OR
1393 * BLWP @PRINTN
1394 *
1395 * STATISTICS: WORKSPACE = IOWKS (SHARED WITH I/O ROUTINES)
1396 0668 PRINT
1397 0668 05EC' DATA IOWKS, PRTCR TRANSFER VECTOR
    066A 0674'
1398 066C PRINTN
1399 066C 05EC' DATA IOWKS, PRTNCR TRANSFER VECTOR
    066E 0670'
1400 0670 PRTNCR
1401 0670 0704 SETO FLGS SET FLAG FOR PRINT WITHOUT CR
1402 0672 1001 JMP PRENT
1403 0674 PRTCR
1404 0674 04C4 CLR FLGS SET FLAG FOR PRINT WITH CR
1405 0676 PRENT
1406 0676 C0E0 MOV @PRLUND, IOPLUN STORE LUND
    0678 0010'
1407 067A PRWRT
1408 067A D0C1 MOVB WRCODP, IOPLUN STORE I/O OP
1409 067C C30D MOV R13, R12 GET CALLER'S WP
1410 067E 022C AI R12, R10+R10 INDEX TO CALLING PARAMETERS
    0680 0014
1411 0682 C15C MOV *R12, BUFADR SET UP BUFFER ADDR
1412 0684 PRTKEY
1413 0684 06A0 BL @GETCNT GET CHAR COUNT
    0686 069C'
1414 0688 A1C4 A FLGS, CHRCNT ADJUST CHRCNT FOR CR
1415 068A 04C4 CLR FLGS INIT FLAGS
1416 068C 04C2 CLR IOC INIT CODE FOR I/O SUPV CALL
1417 068E 0419 BLWP *SVCAL2 MAKE SUPV CALL
1418 0690 A707 A CHRCNT, *R12 INCREMENT PRINT BUFF TO END
1419 0692 C082 MOV IOC, IOC CHECK RETURN STATUS
1420 0694 1302 JEQ PREXT ZERO, EXIT
1421 0696 0460 B @ASK ESC CNTL RETURN OR ERROR
    0698 013C'
1422 069A PREXT
1423 069A 0380 RTWP RETURN

```

```
1425      * TITLE:      GETCNT
1426      *              GET CHARACTER COUNT
1427      * REVISION:   03/15/76
1428      *              ORIGINAL
1429      * COMPUTER:   990, ASSEMBLY
1430      * ABSTRACT:   SCANS LINE COUNTING CHARACTERS UP TO
1431      *              AND INCLUDING A CARRIAGE RETURN.
1432      * CALLING SEQUENCE:
1433      *              R10 = BUFFER ADDRESS
1434      *              BL  @GETCNT
1435      * RETURN PARAMETERS:
1436      *              CHRCNT(R7) = CHARACTER COUNT
1437      * STATISTICS:  WORKSPACE = IOWKS
1438 069C      GETCNT
1439 069C 04C7      CLR  CHRCNT          INIT CHAR COUNT
1440 069E      GCLOOP
1441 069E 0587      INC  CHRCNT          INCR CHAR COUNT
1442 06A0 9835      CB   *BUFADR+, @CR  COMPARE CHAR TO CR
1443      06A2 0007'
1443 06A4 1303      JEQ  GCEXT          IF EQUAL, EXIT
1444 06A6 0287      CI   CHRCNT, 80    MORE THAN 80 CHARS?
1445      06A8 0050
1445 06AA 11F9      JLT  GCLOOP          NO, CONTINUE
1446 06AC      GCEXT
1447 06AC 6147      S    CHRCNT, BUFADR  RESET BUFFER ADDR TO START
1448 06AE 045B      ENDLAL B  *RTN      RETURN
1449      END
```

NO ERRORS

FXRLAL		SDSMAC 947075 *B 16:59:48 SATURDAY, DEC 11, 1976.										
LABEL	VALUE	DEFN	REFERENCES	948926-9901				**	PAGE 0041			
#	R	06B0	0603	0838	0869	0870	0938	1071	1310	1335	1354	
A			0633	0651	0667	0680	0695	0701	0708	0752	0817	
			0819	0839	1062	1366	1414	1418				
AI			0567	0616	0642	0644	0687	0723	0730	0766	0775	
			0804	0871	0940	0979	0991	1006	1314	1359	1410	
ANDI			0408	0557	0563	0965	1369					
ASCVL	R	0538	1097	1141	1154							
ASK	R	013C	0390	0385	0459	0558	0627	1421				
B			0573	0611	0627	0861	1421	1448				
BATCH	R	017C	0413	0415								
BGLAL	D	0000	0179	0047								
BHCODE		0007	0112	1138								
BIAS		0002	0070	0563	0622	0651	0652	0660	0673	0745	0810	
BIASWD	R	01EA	0473	0368	0848							
BIHEX	R	0554	1135	1134								
BINH	R	0564	1143	1145								
BINHEX	R	0550	1132	0450	0758	0999						
BINVAL		0000	0108	1136	1137	1137	1161					
BINVL	R	0530	1093									
BKOUT	R	02BE	0649	0588								
BL			0349	0354	0363	0392	0610	0631	0665	0678	0750	
			0767	0815	0837	0847	0955	1002	1079	1413		
BLANK	R	0008	0188	0634								
BLWP			0336	0353	0362	0380	0382	0396	0399	0412	0414	
			0425	0431	0434	0441	0450	0453	0455	0457	0556	
			0625	0685	0690	0758	0770	0773	0782	0786	0799	
			0868	0937	0947	0964	0996	0999	1005	1070	1081	
			1083	1139	1159	1317	1365	1417				
BSS			0250	1292	1296							
BUFADR		0005	0141	1360	1376	1411	1442	1447				
BUFLEN		0006	0142	1361								
BYTE			0188	0191	0192	0193	0194	0195	0220	0225	0228	
			0233	0238	0245	0251	0255	0257	0261	0263	0267	
			0269	0273	0275							
C			0423	0733	0830	0832	1063	1073	1264	1370		
CB			0384	0400	0402	0458	0634	0967	1270	1442		
CHLP	R	0226	0562	0617	0645	0668	0689	0704	0714			
CHLP2	R	039E	0783	0793	0805	0820	0840	0849				
CHLP3	R	031E	0713	0734	0784							
CHOPC		000E	0117	1136	1140	1152	1160					
CHRCNT		0007	0143	1363	1366	1414	1418	1439	1441	1444	1447	
CI			0570	0764	1444							
CKCOM	R	04FA	1058	1064								
CKL		0000	0097	1069								
CKLOC	R	050C	1071	1014								
CKSM		2306	0174	1078								
CKSYMB	R	0186	0422	0444								
CLR			0383	0398	0432	0549	0564	0565	0650	0791	0975	
			0990	1056	1059	1311	1312	1362	1363	1364	1377	
			1404	1415	1416	1439						
CMBG	R	04F6	1055	1051								
CMWP	R	04D2	1013	1050								
COLON	R	0009	0191	0967								
COMPARE	R	05DA	1269	1273								
CONLOC	R	0410	0870	0866								
CONVL	R	040E	0869	0867								
CONVRT	R	0402	0865	0631	0847	0855						
CONVNS	R	0530	1092	1133	1149							
CR	R	0007	0187	1442								
CSUM		0007	0104	1056	1062	1065	1073					

LABEL	VALUE	DEFN	REFERENCES	948926-9901	**							
CTLUND R	0012	0207	0404	0409	0411	1082	1336					
CUMCHK R	04F2	1049	0799									
CUMPT	0003	0100	1057	1060	1063	1069						
DATA			0180	0181	0185	0205	0206	0207	0217	0219	0227	
			0243	0244	0282	0283	0284	0285	0286	0288	0289	
			0291	0292	0293	0294	0295	0296	0297	0298	0299	
			0340	0451	0452	0471	0472	0473	0475	0476	0477	
			0478	0479	0480	0481	0482	0483	0484	0485	0486	
			0487	0488	0489	0490	0545	0546	0578	0579	0580	
			0581	0582	0583	0584	0585	0586	0587	0588	0589	
			0590	0591	0592	0593	0594	0595	0596	0597	0598	
			0599	0600	0601	0602	0632	0666	0679	0751	0759	
			0760	0816	0838	0848	0869	0870	0879	0880	0881	
			0882	0883	0884	0885	0886	0887	0888	0889	0890	
			0891	0892	0893	0894	0928	0929	0938	0939	1000	
			1001	1014	1015	1016	1017	1018	1019	1020	1021	
			1022	1023	1024	1025	1026	1027	1028	1029	1050	
			1051	1071	1072	1093	1094	1095	1096	1097	1098	
			1099	1100	1101	1102	1103	1104	1105	1106	1107	
			1108	1133	1134	1149	1150	1166	1167	1168	1169	
			1170	1171	1172	1173	1174	1175	1176	1177	1178	
			1179	1180	1181	1207	1208	1216	1217	1227	1228	
			1252	1253	1290	1291	1293	1294	1295	1310	1335	
			1354	1397	1399							
DEC			0372	0426	0692	0776	0949	0982	1272			
DECT			1367									
DEF			0047	0048	0049							
DEFL R	002A	0219	0760									
DEFM R	0014	0216	0769									
DEFN R	0022	0217	0761	0764								
DLDBI	00A0	0212	0366									
DLDPT	0000	0211	0357									
DSBG R	0210	0547	0546									
DSPEXT R	0206	0653	0626									
DSPTCH R	020C	0544	0414									
DSWP R	01E6	0470	0548	1018								
DSWP2 R	01EC	0474	0545	0654								
END R	000C	0194	0400	1449								
ENDACT R	0072	0255	0454									
ENDBUF E		0290	0291	0370								
ENDLAL D	06AE	1448	0048									
ENDLPR R	030A	0700	0697									
ENDOB	0004	0101	1063									
ENDPRO R	0182	0419	0401									
ENDST R	059C	1176	0284	0732	0733	0830	0977					
ENTADD R	00A8	0287	0666	0667								
ENTB R	01C0	0448	0424									
ENTERD R	0322	0720	0774									
ENTERR R	0320	0718	0686									
ENTMSG R	0063	0249	0286									
ENTPT	0004	0057	0449									
ENTVAL	0005	0058	0369	0463								
ENTVL R	00A8	0288	0451									
EOF	2000	0189	0557	0965								
EOU			0053	0054	0055	0056	0057	0058	0059	0060	0061	
			0062	0063	0064	0068	0069	0070	0071	0072	0073	
			0074	0075	0076	0077	0078	0079	0080	0084	0085	
			0086	0087	0088	0089	0090	0091	0092	0093	0097	
			0098	0099	0100	0101	0102	0103	0104	0108	0109	
			0110	0111	0112	0113	0114	0115	0116	0117	0121	

			0122	0123	0124	0125	0126	0127	0128	0129	0130
			0131	0132	0136	0137	0138	0139	0140	0141	0142
			0143	0144	0145	0146	0150	0151	0152	0153	0154
			0155	0156	0157	0158	0159	0160	0161	0162	0163
			0164	0165	0169	0170	0171	0172	0173	0174	0186
			0187	0189	0190	0199	0200	0211	0212	0603	
ERR	R	027A	0609	0640	0740	0861					
ERROR	E		0045	0610	0955	1079					
ESTPT		0002	0055	0371	0423						
EVEN				0201	0276						
EXIT	R	05E6	1276	1265							
F	R	000A	0192	0384							
FIRST	R	047E	0974	0945							
FLAGS		0002	0123	0296	0296	0692	0692	0776	0776	0949	0949 1210
				1274							
FLCL	R	0468	0960	0948							
FLGS		0004	0140	1362	1368	1377	1401	1404	1414	1415	
FLGVL		0007	0075	0564	0719	0722					
FLRT	R	047C	0969	0966							
FST		0001	0085	0944	0944	0975					
FSTIM	R	041A	0880	0374							
FXRF	R	0312	0705	0693							
GCEXT	R	06AC	1446	1443							
GCLOOP	R	069E	1440	1445							
GETCNT	R	069C	1438	1413							
GETHEX	E		0045	0354	0363						
GETSFL	R	05A8	1206	0425							
GETSNM	R	05BC	1226	0434							
GETSVL	R	05B0	1215	0690	0786	0996					
GTSFL	R	05AC	1209	1208							
GTSNM	R	05C0	1229	1228							
GTSVL	R	05B4	1218	1217							
HBCODE		0008	0113	1153							
HCNT		000B	0115	1142	1155						
HEXB	R	057A	1156	1158							
HEXBI	R	0570	1151	1150							
HEXBIN	R	056C	1148	0868	0891	1019					
HEXP		0005	0102	1070							
HEXP2		000C	0093	0937							
HLD		0006	0074	0565	0660	0667	0673	0680	0696	0698	0709 0711
				0745	0752	0810	0817	0825	0839		
HVAL		0001	0098	1059	1060	1061	1062				
IDT				0041							
ILCD		2302	0170	0608							
ILSQ		2301	0169	0860							
INC				0374	0386	0628	0652	0719	0932	1144	1157 1441
INCHK	R	03F6	0853	0485							
INCP		000D	0078	0665	0678	0750	0815	0837			
INCT				0956	0988	1275					
INER	R	03FA	0859	0846							
INIT	E		0045								
INPT		0000	0068	0554	0555	0566	0616	0634	0642	0644	0684 0687
				0724	0725	0726	0727	0728	0729	0763	0766 0772
				0775	0804	0867	0871	1018	1018		
INSID		000E	0079	0549	0623	0623	0628	0650	0845	0845	0854 0854
IOC		0002	0138	1312	1364	1416	1419	1419			
IOERR		4000	0190	0557	0965						
IOPARM		000A	0063	0352	0361	0379	0381	0395	0397	0398	0411 0430
				0435	0437	0438	0439	0440	0449	0454	0456 0555
				0608	0639	0739	0769	0781	0860	0954	0963 1004

Label	Value	Defn	References	948926-9901	**							
			1078	1080	1082							
IOPLUN	0003	0139	1311	1316	1336	1355	1357	1369	1370	1374	1375	
			1406	1408								
IOWKS	R 05EC	1289	1310	1335	1354	1397	1399					
JEQ			0401	0407	0429	0635	0689	0697	0757	0777	0793	
			0950	1074	1371	1420	1443					
JGT			0571	0983								
JHE			0831	0833								
JL			0734									
JLE			0424	1265								
JLT			0427	0568	0693	0765	1064	1145	1158	1445		
JMP			0355	0356	0364	0365	0415	0444	0617	0626	0629	
			0640	0645	0668	0686	0699	0704	0714	0740	0774	
			0784	0794	0805	0820	0840	0849	0948	1283	1337	
			1378	1402								
JMPTB	R 0244	0577	0570	0572								
JMPTBE	R 0276	0603	0570									
JNE			0385	0403	0459	0558	0624	0712	0846	0855	0945	
			0966	0968	1271	1273						
K7	R 0594	1172	0404									
KBLUNO	R 000E	0205	1355	1370								
KEYIN	R 062C	1353	0382	0399	0457	1081						
KEYPLF	R 065A	1373	1371									
KEYRD	R 0634	1356	1337									
LAER	2304	0172	0739									
LAL	R 0008	0341	0340	0611								
LALCSR	D 000E	0335	0049	0180								
LBM5G	R 0094	0273	0361									
LDADER	R 0348	0738	0831									
LDBG	R 043C	0930	0929									
LDBUF	E	0045	0283	0554	0879	1016						
LDPT	000C	0077	0695	0701	0708	0819						
LDR	R 0438	0927	0625									
LDWP	R 0418	0878	0928									
LEN	0008	0144	1361									
LF	R 0006	0186										
LFCR	R 0006	0185	0186	0187	1376							
L1			0344	0346	0352	0357	0361	0366	0370	0375	0379	
			0395	0430	0436	0440	0454	0554	0608	0639	0739	
			0761	0769	0781	0860	0954	0976	0992	1078	1084	
			1141	1154	1376							
L1ST	R 053C	1099	0383	0386	0756							
LOAD	R 000B	0193	0402									
LOADPT	R 01FE	0484	0359									
LOADV2	R 03E4	0838	0836									
LOC	0006	0090	0998									
LOCNUM	R 006C	0250	0452									
LOCPT	0004	0072	0688	0695	0696	0698	0701	0702	0708	0709	0710	
			0711	0788	0792	0792						
LOCPTR	R 0424	0885	0622	1000								
LOCVAL	R 005E	0244	1001									
LODM5G	R 0045	0237	0395									
LP	R 0492	0980	0983									
LPDF	R 0314	0707	0712	0794								
LPFL	R 046A	0962	0968									
LPPXR	R 0300	0694	0699									
LPMSG	R 008B	0267	0352									
LSTMSG	R 007F	0261	0379									
LWPT			0342	0391	0548	0654						
MADD	0002	0110	1140	1143	1152	1156						

R0		0000	0150										
R1		0001	0151										
R10		000A	0160	0357	0359	0366	0368	0369	1314	1314	1359	1359	
				1410	1410								
R11		000B	0161										
R12		000C	0162	1313	1314	1315	1315	1316	1358	1359	1360	1366	
				1367	1368	1409	1410	1411	1418				
R13		000D	0163	1313	1358	1409							
R14		000E	0164	0956	1006								
R15		000F	0165										
R2		0002	0152										
R3		0003	0153										
R4		0004	0154										
R5		0005	0155										
R6		0006	0156										
R7		0007	0157										
R8		0008	0158										
R9		0009	0159	0344	0345	0346	0347	0557	0965				
RD		0000	0084	0931	0961	0963							
RDCOD		0900	0199	1290									
RDCODP		0000	0136	1357									
RDRBF		0001	0054	0381	0384	0397	0400	0402	0405	0406	0456	0458	
RDRL		0002	0099	1057	1080								
READ	R	0622	1334	0556	0964								
REF				0045	0046	0290							
REGLE	R	0146	0394	0403									
RLRG		0006	0103	1073									
RLVL	R	04DE	1020	1072									
RQLB	R	00F8	0360	0364									
RQLP	R	00E2	0351	0355									
RT				0337	0872								
RTN		000B	0146	1448									
RTWP				0464	0655	0970	1007	1086	1146	1162	1211	1222	
				1232	1277	1318	1372	1423					
				0443	0977	1220	1231	1267	1282	1447			
S													
SCBLK	R	05F0	1292	1295									
SOCODE		0003	0111	1138	1153								
SDF	R	03A0	0785	0777									
SETLP	R	03DE	0835	0833									
SETO				0420	1401								
SEVEN		0006	0127	1231	1267								
SFLAGS		000C	0064	0426									
SOC				1316									
SPRNT	R	01A0	0433	0429									
SPTR		0007	0060	0370	0371	0372	0373	0421					
SRC	R	05CA	1257	1253									
SRCSYM	R	05C6	1251	0685	0773	0886							
SRL				0569									
SRSM		0007	0091	0947									
STLB	R	010A	0367	0365									
STLP	R	00F4	0358	0356									
STPTR		0007	0128	1258									
SVCAL1		0009	0114	1139	1159								
SVCAL2		0009	0145	1317	1365	1417							
SVCALT	E		0045	1102	1294								
SVCBLK	R	0536	1096	1103									
SVC5R	E		0046	0346									
SVCWP	E		0046	0344									
SVPT		0005	0073	0691	0702	0703	0706	0721	0722	0723	0724	0725	
				0726	0727	0728	0729	0730	0731	0732	0787	0788	

F	L	H	L	VALUE	DEFN	REFERENCES	948926-9901	**						
						0789	0790	0791						
						0405	1061							
						0978	0979	0981	0989	0990	0991	0993	0994	0995
						0997	0998							
						0285	0296	0435	0691	0692	0721	0776	0787	0790
						0949	0989	0997	1207	1216	1227	1252		
						1283								
						0435	0435	0691	0691	0787	0787	0997	0997	1219
						1220	1221	1230	1231	1266	1267	1270		
						1268	1272							
						0373	0421	0423	0443					
						0285	0285	0721	0721	0790	0790	0989	0989	1210
						1219	1230	1258	1264	1266	1274	1282		
						0291	0978	1173						
						0621	0578							
						0630	0624							
						0641	0635							
						0643	0629							
						0664	0579							
						0659	0580							
						0672	0581							
						0677	0582							
						0744	0583							
						0749	0584							
						0798	0585							
						0803	0586							
						0814	0587							
						0809	0595							
						0829	0596							
						0824	0597							
						0844	0598							
						0568	0571	0589	0590	0591	0592	0593	0594	0599
						0600	1084							
						0615	0601	0602						
						0733	0832	0834						
						0406	0408	0409	0436	0437	0438	0439		
						0443	0977							
						0458								
						0216	0218	0224	0226	0232	0237	0242	0249	
						0256	0262	0268	0274					
						0566	0567	0569	0570	0572	0572	0573	0756	0761
						0763	0764							
						0001	0027	0277	0465	0873	1008	1087	1163	1233
						1284								
						0976	0982	0992	0993	0994	0995			
						1282								
						0487								
						0375								
						0440								
						0436								
						0430								
						0420	0428	0428	0432					
						0463								
						1141	1143	1154	1156	1160	1161			
						1221								
						1291								
						1375	1408							

PXREDT

MIRA990

V2L1

12:13:11

258/76

PAGE 0001A

PXREDT - TEXT EDITOR

948927-9901**

```
0002      * TITLE.      PXREDT
0004      *              TEXT EDITOR
0005      * REVISION:
0006      *              ORIGINAL
0007      *              03/15/76
0008      *              MODIFIED TO RUN WITH PX9MTR, THE PROTOTYPING
0009      *              SYSTEM AND 733ASR PROGRAM DEVELOPMENT SYSTEM
0010      *              DEBUG MONITOR
0011      *              06/28/76
0012      *              MODIFIED FOR INCLUSION IN MONITOR
0013      *              ALONG WITH THE TEXT EDITOR.
0014      * COMPUTER: 990, 990 ASSEMBLY -MIRA ASSEMBLER USED
0015      * ABSTRACT: PXREDT IS AN INTERACTIVE PROGRAM FOR EDITING
0016      *              SOURCE OR OBJECT CODE. THE USER MAY GENERATE
0017      *              NEW TEXT DATA FROM THE KEYBOARD OR MAY READ
0018      *              IN EXISTING TEXT DATA FROM CASSETTE. THE
0019      *              TEXT DATA, NEW OR EXISTING, IS READ INTO A
0020      *              BUFFER AREA AND MAY BE EDITED BY ADDITIONS,
0021      *              DELETIONS, OR CORRECTIONS FROM THE KEYBOARD.
0022      *              THE EDITED DATA MAY THEN BE WRITTEN TO
0023      *              CASSETTE TAPE.
0024      *
0025      *              PROGRAM FUNCTIONS:
0026      *              1) SET UP COMMANDS
0027      *              2) POINTER COMMANDS
0028      *              3) EDIT COMMANDS
0029      *              4) PRINT COMMANDS
0030      *              5) OUTPUT COMMANDS
```

```

0033 * TITLE: DATA AREA
0034 * REVISION:
0035 * ORIGINAL
0036 * 03/15/76
0037 * MODIFIED TO RUN WITH PXRMTN
0038 * COMPUTER: 990, 990 ASSEMBLY
0039 * ABSTRACT: THE DATA AREA CONTAINS THE WORKSPACES,
0040 * BUFFERS, AND MISCELLANEOUS DATA FOR PXREDT
0041 * CALLING SEQUENCE: NON-CALLABLE
0042 *
0043 IDT 'PXREDT'
0044 0000 RORG
0045 *
0046 * DX10 TASK INITIATION VALUES
0047 *
0048 0000 0006' DATA MAINRG WORKSPACE
0049 0002 01A6' DATA START ENTRY POINT
0050 0004 021C' DATA ESCRTN END ACTION PC
0051 *
0052 * WORKSPACE AREA FOR MAIN DRIVER
0053 *
0054 MAINRG
0055 0006 1040' DATA INIT R0 ENTRY
0056 REF ENDBUF DEFINE END OF BUFFER AREA
0057 0008 0000 HICOR DATA ENDBUF R1 TOPMEM
0058 000A 0317' DATA MSG01 R2 ACTION
0059 000C 03B0' DATA MSG12 R3 PROMPT
0060 000E 0FEC' DATA PRINT R4 PRT
0061 0010 0FF0' DATA PRINTN R5 PRTN
0062 0012 0FA4' DATA KEYIN R6 KEY
0063 0014 BSS 18 R7-R15
0064 *
0065 * MAIN DRIVER WORKSPACE EQUATES
0066 *
0067 0000 ENTRY EQU 0
0068 0001 TOPMEM EQU 1
0069 0002 ACTION EQU 2
0070 0003 PROMPT EQU 3
0071 0004 PRT EQU 4
0072 0005 PRTN EQU 5
0073 0006 KEY EQU 6
0074 *
0075 * WORKSPACE AREA FOR COMMAND SUBROUTINES
0076 *
0077 0026 00A4 ROUTRG DATA PGEND+4 R0 ADDR OF LINE AFTER DUMMY LINE
0078 0028 BSS 6 R1-R3
0079 002E 00A8 VLNPTR DATA PGEND+8 R4 VALUE FOR LIMPTR
0080 0030 BSS 22 R5-R15
0081 0046 BSS 6 R13-R15 EXTENDED FOR BLWP IN FIND
0082 002C' ROUTR1 EQU ROUTRG+6
0083 *
0084 * COMMAND WORKSPACE EQUATES
0085 *
0086 0000 DUMNXT EQU 0 LINE AFTER DUMMY

```

0087	0001	UDCNT	EQU	1	UP-DOWN COUNT
0088	0002	LINAD	EQU	2	LINE ADDRESS
0089	0003	PLIMTS	EQU	3	PRINT LIMITS
0090	0004	LINPTR	EQU	4	LINE POINTER
0091	0005	CLLOC	EQU	6	CURRENT LINE-NO LOCATION
0092	0007	TMLOC	EQU	7	TEMPORARY LOCATION
0093	0008	CLINNO	EQU	8	CURRENT LINE NUMBER
0094	000B	RTN	EQU	11	RETURN ADDRESS FOR BL
0095	0008	FCHAR	EQU	8	
0096	000D	FGPOS	EQU	13	FIND-TEXT GET POSITION
0097	000E	FPPOS	EQU	14	FIND-TEXT PUT POSITION
0098	000F	NEWPTR	EQU	15	LINE POINTER FOR NEW LINE
0099		*			
0100		* WORKSPACE AREA FOR INPUT/OUTPUT ROUTINES			
0101		*			
0102		IOWKS			
0103	004C	0900	DATA	RDCOD	R0 RDCODP
0104	004E	0B00	DATA	WRCOD	R1 WRCODP
0105	0050		SCBLK	BSS	16 R2-R9 I/O SUPV CALL BLK (PRE)
0106	0060	0050	DATA	00	R10 LEN
0107	0062		BSS	10	R11-R15
0108		*			
0109		* I/O WORKSPACE EQUATES			
0110		*			
0111	0000	RDCODP	EQU	0	
0112	0001	WRCODP	EQU	1	
0113	0002	IOC	EQU	2	
0114	0003	IOPLUN	EQU	3	
0115	0004	FLGS	EQU	4	
0116	0005	BUFADR	EQU	5	
0117	0006	BUFLEN	EQU	6	
0118	0007	CHRCNT	EQU	7	
0119	000A	LEN	EQU	10	
0120		*			
0121		* COMMON REGISTER EQUATES			
0122		*			
0123	0000	R0	EQU	0	
0124	0001	R1	EQU	1	
0125	0002	R2	EQU	2	
0126	0003	R3	EQU	3	
0127	0004	R4	EQU	4	
0128	0005	R5	EQU	5	
0129	0006	R6	EQU	6	
0130	0007	R7	EQU	7	
0131	0008	R8	EQU	8	
0132	0009	R9	EQU	9	
0133	000A	R10	EQU	10	
0134	000B	R11	EQU	11	
0135	000C	R12	EQU	12	
0136	000D	R13	EQU	13	
0137	000E	R14	EQU	14	
0138	000F	R15	EQU	15	

```

0140      *
0141      *  COMMAND BUFFER
0142      *
0143  006C  006E' CPBUFA DATA $+2      COMMAND BUFFER ADDRESS
0144  006E      CPBUF  BSS  82      COMMAND BUFFER
0145      *
0146      *  PRINT BUFFER
0147      *
0148  00C0  00C2' PBUFA  DATA $+2      PRINT BUFFER ADDRESS
0149  00C2      PBUF   BSS  82      PRINT BUFFER
0150      *
0151      *  FLAGS
0152      *
0153  0114      EOFLG  BSS  2      END OF FILE FLAG
                                =0 IF NO EOF HAS BEEN ENCOUNTERE
0154      *                                =1 IF EOF HAS BEEN ENCOUNTERED
0155      *
0156  0116      CHFLG  BSS  2      CHANGE FLAG
                                =0 IF NOT EXECUTING CHANGE CMD
0157      *                                =1 IF EXECUTING CHANGE COMMAND
0158      *
0159  0118      EXTFLG BSS  2      EXIT FLAG
                                =0 IF QUIT HAS NOT BEEN EXECUTED
0160      *                                SET TO 1 BY QUIT CMD
0161      *
0162  011A      PLFLG  BSS  2      PRINT LINE-NO FLAG
                                =0 IF LINE NUMBERS ARE NOT TO BE
0163      *                                =-1 IF LINE NOS ARE TO BE PRINTE
0164      *
0165  011C      ABRLFG BSS  2      ABSOLUTE-RELATIVE FLAG
                                =0 IF ABSOLUTE OPERAND
0166      *                                >0 IF RELATIVE
0167      *
0168  011E      LFFLG  BSS  2      LABEL-FIRST OCCURANCE FLAG
                                =0 FOR FIRST OCCURANCE
0169      *                                =-1 FOR LABEL
0170      *
0171  0120      PFLG   BSS  2      PRINT FLAG
0172  0122      VFLG   BSS  2      VERIFY FLAG
0173  0124      REPFLG BSS  2      STRING REPLACEMENT FLAG
                                =0 FOR NO REPLACEMENT
0174      *                                NOT =0 FOR NO REPLACEMENT
0175      *
0176  0126      NWLF   BSS  2      NEW LINE FLAG
0177  0128  0000  FFLG   DATA 0      FIND FLAG
0178      *
0179      *  DATA DEFINITIONS
0180      *
0181  012A      MALIN  BSS  2      MOVE AFTER LINE
0182  012C      LAMOV  BSS  2      LINE AFTER LINES TO BE MOVED
0183  012E  0001  MARGLM DATA 1, 72  DEFAULT MARGIN LIMITS
0184  0130  0048
0184  0132      PMRGIN BSS  2      RIGHT MARGIN FOR PRINT
0185  0134      LOMRG  BSS  2      LOW MARGIN FOR FIND
0186  0136      HIMRG  BSS  2      HIGH MARGIN FOR FIND
0187  0138      CC     BSS  2      CHARACTER COUNT
0188  013A      NUMBF  BSS  2      NUMBER FOUND FLAG FOR OPERAND EXTR
0189  013C      NUMB   BSS  2      OPERAND
0190  013E      SLADDR BSS  2      SQUEEZE-LINE ADDRESS
0191  0140      SQPLAD BSS  2      SQUEEZE-PREVIOUS LINE ADDRESS
0192  0142      SONLAD BSS  2      SQUEEZE-NEXT LINE ADDRESS

```

0193	0144	LINADD	BSS	2	
0194	0146	LINADR	BSS	2	
0195	0148	PRELAD	BSS	2	
0196	014A	NXLAD	BSS	2	
0197	014C	FINLAD	BSS	2	
0198	014E	NXTLAD	BSS	2	
0199	0150	CURNO	BSS	2	
0200	0152	PRLAD	BSS	2	
0201	0154	SAVRTN	BSS	2	LOCATION TO SAVE LINK REGISTER IN
0202	0156	SLOC	BSS	2	
0203	0158	SLINPT	BSS	2	
0204	015A	LML	BSS	2	LINE NUMBER OF LAST LINE MOVED
0205	015C	MAXLIN	BSS	2	LINE NUMBER OF LAST SOURCE LINE BR
0206	015E	PRVLIN	BSS	2	
0207	0160	SVPRM	BSS	2	LOCATIONS IN WHICH MARGINS
0208	0162	SAVMRG	BSS	2	ARE SAVED
0209	0164	FSD	BSS	2	FIRST STRING DELIMITER
0210	0166	SRKNT	BSS	2	STRING CHARACTER COUNT
0211	0168	SSD	BSS	2	SECOND STRING DELIMITER
0212	016A	FXLAD	BSS	2	
0213	016C	THLAD	BSS	2	
0214	016E	SCFP	BSS	2	
0215	0170	STRCHR	BSS	2	
0216	0172	FSDXT	BSS	2	
0217	0174	FSDX	BSS	2	LOCATION OF FIRST STRING DELIMITER
0218	0176	SSDX	BSS	2	LOCATION OF SECOND STRING DELEMETE
0219	0178	MVRCNT	BSS	2	MOVE RELATIVE COUNT
0220	017A	8000	INLNNO	DATA	>8000
0221	017C	000A	TEN	DATA	10
0222	017E	00A0	ANCHOR	DATA	PGEND
0223	0180	00A4	INTLPT	DATA	PGEND+4
0224	0182	270F	LLIMIT	DATA	9999
0225	0184	0A0D	LFCR	DATA	>0A0D
0226		0184	LF	EQU	LFCR
0227		0185	CR	EQU	LFCR+1
0228	0186	200D	SPCR	DATA	>200D
0229		0186	SP	EQU	SPCR
0230	0188	0000	PRTPOS	DATA	0
0231	018A	0007	TABS	DATA	7, 12, 30, 32, 0
	018C	000C			
	018E	001E			
	0190	0020			
	0192	0000			
0232		0007	BELL	EQU	>07
0233		000A	LFE	EQU	>0A
0234		000D	CRE	EQU	>0D
0235		2000	EOF	EQU	>2000
0236		4000	IOERR	EQU	>4000
0237	0194	30	BINASC	BYTE	>30
0238	0195	59	YEA	BYTE	'Y'
0239	0196	43	CONT	BYTE	'C'
0240	0197	54	TERM	BYTE	'T'
0241	0198	09	TAB	BYTE	>09


```
0243      *
0244      * PERIPHERAL DEVICE DEFAULT LOGICAL UNIT NUMBERS
0245      *
0246  019A  0000  KBLUND DATA 0          LOG (KEYBOARD)
0247  019C  0006  PRLUND DATA 6          LOG (PRINTER)
0248  019E  0007  C1LUND DATA 7          CS1 (LEFT CASSETTE - INPUT)
0249  01A0  0008  C2LUND DATA 8          CS2 (RIGHT CASSETTE - OUTPUT)
0250      *
0251      * NON I/O MONITOR SUPV CALL BLOCKS
0252      *
0253  01A2  0400  ENDPRG DATA >0400          END OF PROGRAM
0254      *
0255      * MONITOR SUPV CALL CODES FOR I/O
0256      *
0257  01A4    00  OPNCOD BYTE >00          OPEN
0258  01A5    00  EOF COD BYTE >0D          END OF FILE
0259    0900  RDCOD EQU >0900          READ ASCII
0260    0B00  WRCOD EQU >0B00          WRITE ASCII
0261      EVEN
0262      *
0263      * DEFINE SUPV CALL XOP
0264      *
0265      DXOP SVC.15
```

```

0268 * TITLE: START/PXRREDT
0269 *
0270 * REVISION:
0271 * ORIGINAL
0272 * 03/15/76
0273 *
0274 * MODIFIED TO RUN WITH PXRMTN
0275 * COMPUTER: 990, 990 ASSEMBLY
0276 * ABSTRACT: THIS IS THE MAIN DRIVER FOR PXRREDT.
0277 * THE FIRST TIME PXRREDT IS EXECUTED AFTER
0278 * BEING LOADED, START WILL BRANCH TO
0279 * INITIALIZATION SECTION FOR HEADING AND
0280 * MEMORY SIZE INIT, IT WILL RETURN TO PXRREDT
0281 * UPON COMPLETION.
0282 * SUBSEQUENT EXECUTIONS WILL FALL THROUGH
0283 * TO PXRREDT, THIS CONTAINS INITIALIZATION
0284 * OF FLAGS, POINTERS, AND BUFFERS AND THE
0285 * MAIN DRIVER LOOP WHICH READS THE
0286 * COMMAND LINE AND BRANCHES TO THE
0287 * APPROPRIATE FUNCTION
0288 DEF EDTCSR
0289 EDTCSR
0290 START
0291 01A6 02E0 LWPI MAINRG INIT WORKSPACE
0292 01A8 0006
0293 01AA 06A0 BL @INIT INITIALIZE PROGRAM
0294 01AC 1040
0295 PXRREDT
0296 01AE 02E0 LWPI MAINRG
0297 01B0 0006
0298 01B2 0300 LIM1 2 INIT STATUS
0299 01B4 0002
0300 01B6 04E0 CLR @MAXLIN INITIALIZE FLAGS & VARIABLES
0301 01B8 015C
0302 01BA 04E0 CLR @EOFLG INIT END OF FILE FLAG
0303 01BC 0114
0304 01BE 04E0 CLR @EXTFLG INIT FLAG FOR EXIT
0305 01C0 0118
0306 01C2 0720 SET0 @PLFLG SET FLAG TO PRINT LINE #'S
0307 01C4 011A
0308 01C6 C820 MOV @ANCHOR,@ROUTRG+12 SET INITIAL POINTER VALUES
0309 01C8 017E
0310 01CA 0032
0311 01CC 04E0 CLR @ROUTRG+16 SET UP INITIAL LINNO.
0312 01CE 0036
0313 01D0 C820 MOV @ANCHOR,@VLNPTR SET UP POINTER VALUES
0314 01D2 017E
0315 01D4 002E
0316 01D6 020A LI R10,8
0317 01D8 0008
0318 01DA A80A A R10,@VLNPTR INCREMENT PAST DUMMY LINE
0319 01DC 002E
0320 01DE 091A SRL R10,1
0321 01E0 04EA CLR DUM CLR @PGEND(R10) CLEAR DUMMY LINE
0322 01E2 00A0

```

```

0306 01E4 064A      DECT R10
0307 01E6 18FC      JOC CLRDM
0308 01E8 C820      MOV @SPCR,@PGEND+6      SET UP DUMMY LINE
      01EA 0186'
      01EC 00A6
0309 01EE C820      MOV @MARGLM,@LOMRG      SET LOW FIND MARGIN
      01F0 012E'
      01F2 0134'
0310 01F4 C820      MOV @MARGLM+2,@HIMRG      SET HIGH FIND MARGIN
      01F6 0130'
      01F8 0136'
0311 01FA C820      MOV @MARGLM+2,@PMRGIN      SET RIGHT MARGIN FOR PRINT
      01FC 0130'
      01FE 0132'
0312 0200 06A0      BL @MSGOUT      PRINT LOAD INSTRUCTIONS
      0202 02CE'
0313 0204 0000      DATA 0
0314 0206 06A0      BL @GETANS      WAIT FOR RESPONSE
      0208 02DA'
0315 020A C2A0      MOV @C1LUND,R10
      020C 019E'
0316 020E 0420      BLWP @OPEN      OPEN CASSETTE 1
      0210 0F68'
0317 0212 C2A0      MOV @C2LUND,R10
      0214 01A0'
0318 0216 0420      BLWP @OPEN      OPEN CASSETTE 2
      0218 0F68'
0319 021A 1005      JMP COMMND
0320
0321 *
0322 * ESCAPE CONTROL RETURN ENTRY POINT
0323 *
0324 *
      ESCRTN
0324 021C 02E0      LWPI MAINRG
      021E 0006'
0325 0220 020A      LI R10,LFCR      PRINT LINE FEED/CARRIAGE RETU
      0222 0184'
0326 0224 0414      BLWP *PRT
0327 *
      COMMND
0328 0226 02E0      LWPI MAINRG      ENTRY PNT FOR ESC CONTROL RTN
      0228 0006'
0329 022A 04E0      CLR @CHFLG      CLEAR CHANGE FLAG
      022C 0116'
0330 022E 04E0      CLR @FFLG      CLEAR FIND FLAG
      0230 0128'
0331 *
      GETCOM
0332 0232 C283      MOV PROMPT,R10      PRINT ' ? '
0333 0234 0415      BLWP *PRTN
0334 0236 C2A0      MOV @CPBUFA,R10
      0238 006C'
0335 023A 0416      BLWP *KEY      INPUT COMMAND FROM KEYBOARD
0336 023C 06A0      MOVE @CR,*R10      END STRING WITH CR
      023E 0185'
0337 0240 02A0      C @CPBUFA,R10      ANY INPUT?
      0242 006C'
0338 0244 13F6      JEQ GETCOM

```

```

0339          * THIS ROUTINE DETERMINES WHICH COMMAND
0340          * HAS BEEN SPECIFIED AND BRANCHES TO
0341          * THE APPROPRIATE COMMAND SUBROUTINE
0342          *
0343 0246 020A          LI R10, ENDLST-CMDLST
          0248 000E
0344 024A 982A  CMPR  CB  @CMDLST(R10), @CPBUF  DETERMINE COMMAND
          024C 02A0 ✓
          024E 006E ✓
0345 0250 1304          JEQ  SUCCES          FOUND
0346 0252 060A          DEC  R10          IS IT ILLEGAL?
0347 0254 18FA          JOC  CMPR          NO, NOT YET
0348 0256 0460          B    @OPERR          YES
          0258 03FE ✓
0349 025A A28A  SUCCES  A    R10, R10
0350 025C C2AA          MOV  @ROUTAD(R10), R10  GET ADDRESS OF DESIRED ROUTNE
          025E 02B0 ✓
0351 0260 04E0          CLR  @CC
          0262 0138 ✓
0352 0264 05A0          INC  @CC          SET CHARACTER COUNT TO ONE
          0266 0138 ✓
0353 0268 041A          BLWP *R10          EXECUTE COMMAND
0354 026A C320          MOV  @EXTFLG, R12      WAS IT QUIT?
          026C 0118 ✓
0355 026E 13DB          JEQ  COMMND          NO
0356 0270 0420          BLWP @ENDFIL          QUIT -- WRITE END OF FILE
          0272 0F84 ✓
0357 0274 06A0          BL   @MSGOUT
          0276 02CE ✓
0358 0278 0014          DATA 20          PRINT END EDIT MESSAGE
0359          ENDACT
0360 027A C282          MOV  ACTION, R10      TERMINATE/CONTINUE? MSG
0361 027C 0415          BLWP *PRTN
0362 027E 06A0          BL   @GETANS          INPUT RESPONSE
          0280 02DA ✓
0363 0282 028A          CI   R10, PBUF          ANY RESPONSE?
          0284 00C2 ✓
0364 0286 13F9          JEQ  ENDACT          NO, ASK AGAIN
0365 0288 9820          CB   @PBUF, @TERM     TERMINATE?
          028A 00C2 ✓
          028C 0197 ✓
0366 028E 1306          JEQ  EPEXIT          YES, EXIT
0367 0290 9820          CB   @PBUF, @CONT     CONTINUE?
          0292 00C2 ✓
          0294 0196 ✓
0368 0296 16F1          JNE  ENDACT          NO, ASK AGAIN
0369 0298 0460          B    @PXREDT          NO, RESTART EDIT
          029A 01AE ✓
0370 029C 2FE0  EPEXIT  XOP  @ENDPRG, 15      RETURN TO MONITOR
          029E 01A2 ✓

```

```
0372 *  
0373 * LIST OF COMMANDS  
0374 *  
0375 02A0 42 CMDLST BYTE 'B' BOTTOM  
0376 02A1 43 BYTE 'C' CHANGE  
0377 02A2 44 BYTE 'D' DOWN  
0378 02A3 45 BYTE 'E' EXIT  
0379 02A4 46 BYTE 'F' FIND  
0380 02A5 49 BYTE 'I' INSERT  
0381 02A6 48 BYTE 'K' KEEP  
0382 02A7 4C BYTE 'L' LIMITS  
0383 02A8 4D BYTE 'M' MOVE  
0384 02A9 50 BYTE 'P' PRINT  
0385 02AA 51 BYTE 'Q' QUIT  
0386 02AB 52 BYTE 'R' REMOVE  
0387 02AC 53 BYTE 'S' SETUP  
0388 02AD 54 BYTE 'T' TOP  
0389 02AE 55 ENDLST BYTE 'U' UP
```

```
0390 *  
0391 * LIST OF COMMAND ROUTINE ADDRESSES  
0392 *  
0393 02B0 0640' ROUTAD DATA BTM BOTTOM  
0394 02B2 066E' DATA CHNG CHANGE  
0395 02B4 0680' DATA DWN DOWN  
0396 02B6 0794' DATA EXT EXIT  
0397 02B8 0798' DATA FND FIND  
0398 02BA 0A7C' DATA INSRT INSERT  
0399 02BC 0B0A' DATA KP KEEP  
0400 02BE 0B44' DATA LMTS LIMITS  
0401 02C0 0B88' DATA MV MOVE  
0402 02C2 0C38' DATA PRNT PRINT  
0403 02C4 0D4E' DATA QT QUIT  
0404 02C6 0E34' DATA RMV REMOVE  
0405 02C8 0EBE' DATA STP SETUP  
0406 02CA 0F4C' DATA TOP TOP  
0407 02CC 06CA' DATA UP UP
```

```

0410 * TITLE: MSGOUT
0411 * MESSAGE OUTPUT ROUTINE
0412 * REVISION:
0413 * ORIGINAL
0414 * 03/15/76
0415 * MODIFIED TO RUN WITH FXRMTR
0416 * COMPUTER: 990, 990 ASSEMBLY
0417 * ABSTRACT: PRINT MESSAGE SPECIFIED BY N
0418 *
0419 * N MESSAGE
0420 * 0 SETUP INSTRUCTIONS
0421 * 2 TERMINATE/CONTINUE?
0422 * 4 LINE NOT FOUND
0423 * 6 INVALID OPERATOR
0424 * 8 OFF THE TOP
0425 * 10 INVALID OPERAND
0426 * 12 END OF FILE
0427 * 14 0000 FOUND
0428 * 16 LAST LINE
0429 * 18 BUFFER FULL
0430 * 20 END EDIT
0431 * 22 BUFFER EMPTY
0432 *
  
```

```

0433 * CALLING SEQUENCE:
0434 * BL @MSGOUT
0435 * DATA N
0436 *
  
```

```

0437 MSGOUT
0438 020E C2BB MOV *RTN+,R10 INDEX FOR MESSAGE TABLE
0439 02D0 C2AA MOV @MCOUNT(R10),R10 SET UP BUFFER ADDR FOR PRINT
0440 02D4 0420 BLWP @PRINT PRINT MSG
0441 02D6 0FEC'
0442 02D8 045B B *RTN RETURN
  
```

```

0443 * ROUTINE THAT ACCEPTS ANSWERS UP TO A CR
0444 * IN PRINT BUFFER
0445 *
  
```

```

0446 GETANS
0447 02DA C2A0 MOV @PBUFA,R10 SET UP BUFFER ADDRESS
0448 02DC 00C0'
0449 02DE 0420 BLWP @KEYIN INPUT CHARS FROM KBD
0450 02E0 0FA4'
0451 02E2 045B B *RTN
  
```

```

0452 * MESSAGE TABLE
0453 *
0454 02E4 02FC' MOUNT EQU $
0455 02E6 0317' DATA MSG00
0456 02E8 032C' DATA MSG01
0457 02EA 033C' DATA MSG02
0458 02EC 034E' DATA MSG03
0459 02EE 035B' DATA MSG04
  
```

```

0460 02F0 036C' DATA MSG06
0461 02F2 0379' DATA MSG07
0462 02F4 0380' DATA MSG08
0463 02F6 038B' DATA MSG09
0464 02F8 0398' DATA MSG10
0465 02FA 03A2' DATA MSG11
0466
0467 *
0468 * MESSAGES
0469 *
0469 02FC 0A MSG00 BYTE LFE
0470 02FD 50 TEXT 'POSITION TAPES, ENTER CR'
0471 0315 0A BYTE LFE, CRE
0471 0316 0D
0472 0317 54 MSG01 TEXT 'TERMINATE/CONTINUE?'
0473 032A 07 BYTE BELL, CRE
0473 032B 0D
0474 032C 4C MSG02 TEXT 'LINE NOT FOUND'
0475 033A 0A BYTE LFE, CRE
0475 033B 0D
0476 033C 49 MSG03 TEXT 'INVALID OPERATOR'
0477 034C 0A BYTE LFE, CRE
0477 034D 0D
0478 034E 4F MSG04 TEXT 'OFF THE TOP'
0479 0359 0A BYTE LFE, CRE
0479 035A 0D
0480 035B 49 MSG05 TEXT 'INVALID OPERAND'
0481 036A 0A BYTE LFE, CRE
0481 036B 0D
0482 036C 45 MSG06 TEXT 'END OF FILE'
0483 0377 0A BYTE LFE, CRE
0483 0378 0D
0484 0379 46 MSG07 TEXT 'FOUND'
0485 037E 0A BYTE LFE, CRE
0485 037F 0D
0486 0380 4C MSG08 TEXT 'LAST LINE'
0487 0389 0A BYTE LFE, CRE
0487 038A 0D
0488 038B 42 MSG09 TEXT 'BUFFER FULL'
0489 0396 0A BYTE LFE, CRE
0489 0397 0D
0490 0398 45 MSG10 TEXT 'END EDIT'
0491 03A0 0A BYTE LFE, CRE
0491 03A1 0D
0492 03A2 42 MSG11 TEXT 'BUFFER EMPTY'
0493 03AE 0A BYTE LFE, CRE
0493 03AF 0D
0494 03B0 20 MSG12 TEXT ' ? '
0495 03B2 07 BYTE BELL, CRE
0495 03B3 0D
0496 03B4 20 MSG13 TEXT ' Y/N?'
0497 03B9 0D BYTE CRE
0498 03BA 52 MSG14 TEXT 'RDY TAPE-TYPE CR'
0499 03CA 0D BYTE CRE
0500 EVEN
0502 *
  
```

WARNING MESSAGES

948927-9901**

```

0503          *  ROUTINES THAT PRINT WARNING MESSAGES
0504          *
0505 0300 06A0 LSTLIN BL @MSGOUT 'LAST LINE'
      030E 02CE'
0506 03D0 0010          DATA 15
0507 03D2 0450          B @COMMND
      03D4 0226'
0508 03D6 C24B ATTOP MOV RTN, 9
0509 03D8 06A0          BL @MSGOUT 'OFF THE TOP'
      03DA 02CE'
0510 03DC 0008          DATA 8
0511 03DE 0459          B *9
0512 03E0 06A0 OPNERR BL @MSGOUT 'INVALID OPERAND'
      03E2 02CE'
0513 03E4 000A          DATA 10
0514 03E6 0450          B @COMMND
      03E8 0226'
0515 03EA 06A0 MEMPTY BL @MSGOUT 'BUFFER EMPTY'
      03EC 02CE'
0516 03EE 0016          DATA 22
0517 03F0 0450          B @COMMND
      03F2 0226'
0518 03F4 06A0 NOTFND BL @MSGOUT 'LINE NOT FOUND'
      03F6 02CE'
0519 03F8 0004          DATA 4
0520 03FA 0450          B @COMMND
      03FC 0226'
0521 03FE 06A0 OPERR BL @MSGOUT 'INVALID OPERATOR'
      0400 02CE'
0522 0402 0006          DATA 6
0523 0404 0450          B @GETCOM
      0406 0232'

```



```

0526 * TITLE: SQUEEZ
0527 * MEMORY COMPRESS ROUTINE
0528 * REVISION:
0529 * ORIGINAL
0530 * 03/15/76
0531 * MODIFIED TO RUN WITH PXRMT
0532 * COMPUTER: 990, 990 ASSEMBLY
0533 * ABSTRACT: THIS ROUTINE IN EFFECT SQUEEZES OUT INACTIVE
0534 * LINES FROM THE LINE BUFFER BY PLACING ALL
0535 * THE ACTIVE LINES AS ONE CONTIGUOUS BLOCK
0536 * IN MEMORY.
0537 * CALLING SEQUENCE:
0538 * BL @SQUEEZ
0539 * STATISTICS:
0540 * LINPTR POINTS TO THE CELL FOLLOWING THE
0541 * LAST LINE IN MEMORY BEFORE COMPRESSION.
0542 * REGISTER 10 CONTAINS THE ADDRESS OF THE
0543 * CELL FOLLOWING THE LAST LINE AFTER
0544 * COMPRESSION AT LABEL SQ7.
0545 0408 0714 SQUEEZ SETO *LINPTR TERM BUFFER WITH -1
0546 040A C1E0 MOV @INTLPT, TMLOC SET SQUEEZE
040C 0180
0547 040E 0647 DECT TMLOC SEARCH POINTERS
0548 0410 05C7 SQ1 INCT TMLOC
0549 0412 C257 MOV *TMLOC, 9 LOOK FOR INACTIVE BLOCK
0550 0414 0289 CI 9, >FFFF FOUND?
0416 FFFF
0551 0418 16FB JNE SQ1 NO
0552 041A C287 MOV TMLOC, R10 YES
0553 041C 8107 SQ2 C TMLOC, LINPTR IS TO BE MOVED
0554 041E 1450 JHE SQ7 NO MORE TO SQUEEZE
0555 SQ8
0556 0420 0587 INC TMLOC
0557 0422 D257 MOVB *TMLOC, 9
0558 0424 9809 CB R9, @CR WITHIN ACTIVE BLOCK?
0426 0185
0559 0428 16FB JNE SQ8 YES
0560 042A 21E0 COC @MARGLM, TMLOC WORD BOUNDARY?
042C 012E
0561 042E 1601 JNE #+4
0562 0430 0607 DEC TMLOC IF NOT, MAKE IT ONE.
0563 0432 05C7 INCT TMLOC
0564 0434 C257 MOV *TMLOC, 9
0565 0436 0289 CI 9, >FFFF ANOTHER INACTIVE BLOCK?
0438 FFFF
0566 043A 13F0 JEQ SQ2
0567 043C C307 SQ4 MOV TMLOC, 12 SAVE ADDR OF LINE TO BE SQUEEZED U
0568 * SAVE INFO NEEDED TO UPDATE NEXT-LINE PTR OF PREVIOUS
0569 * LINE AND PREVIOUS LINE POINTER OF FOLLOWING LINE
0570 043E 8248 C CLINNO, 9 IS THE LINE AT THE POINTER BEI
0571 0440 1601 JNE #+4
0572 0442 C18A MOV 10, CLLOC YES, ADJUST FOR IT.
0573 0444 808C C 12, LINAD IF LINAD OR PRVLIN
0574 0446 1601 JNE #+4 ARE POINTING TO

```

0575	0448	C08A		MOV	10, LINAD	LINES ABOUT TO BE
0576	044A	880C		C	12, @PRVLIN	MOVED, CORRECT
	044C	015E'				
0577	044E	1608		JNE	SQ4A	THEIR VALUES
0578	0450	C80A		MOV	10, @PRVLIN	
	0452	015E'				
0579	0454	C80A		MOV	10, @SLOC	
	0456	0156'				
0580	0458	05E0		INCT	@SLOC	
	045A	0156'				
0581	045C	05E0		INCT	@SLOC	
	045E	0156'				
0582	0460	C24A	SQ4A	MOV	10, 9	
0583	0462	C809		MOV	9, @SLADDR	
	0464	013E'				
0584	0466	05C9		INCT	9	SAVE POINTERS TO CELLS
0585	0468	C809		MOV	9, @SQPLAD	WHICH WILL CONTAIN
	046A	0140'				
0586	046C	05C9		INCT	9	ADDRESSES OF CELLS TO
0587	046E	C809		MOV	9, @SQNLAD	BE UPDATED
	0470	0142'				
0588			SQ3			
0589	0472	0209		LI	9, 2	
	0474	0002				
0590	0476	CEBC	SQ3A	MOV	*12+, *10+	MOVE THE LINE
0591	0478	0609		DEC	9	UP TO
0592	047A	18FD		JOC	SQ3A	NEW
0593	047C	D6BC	SQ9	MOVB	*12+, *10	LOCATION
0594	047E	983A		CB	*R10+, @CR	STOP MOVING WHEN C/R FOUND
	0480	0185'				
0595	0482	16FC		JNE	SQ9	
0596	0484	C260		MOV	@SQPLAD, 9	UPDATE NEXT-LINE
	0486	0140'				
0597	0488	C259		MOV	*9, 9	POINTER OF
0598	048A	0229		RI	9, 4	PREVIOUS
	048C	0004				
0599	048E	C660		MOV	@SLADDR, *9	LINE
	0490	013E'				
0600	0492	C260		MOV	@SQNLAD, 9	IS THERE A
	0494	0142'				
0601	0496	C259		MOV	*9, 9	NEXT LINE?
0602	0498	1303		JEQ	SQ5	NO
0603	049A	05C9		INCT	9	YES
0604	049C	C660		MOV	@SLADDR, *9	LINE POINTER
	049E	013E'				
0605			SQ5			
0606			*SET SQUEEZE ADDRESS ON WORD BOUNDRY			
0607	04A0	22A0		COC	@MARGLM, 10	
	04A2	012E'				
0608	04A4	1601		JNE	#+4	
0609	04A6	058A		INC	10	
0610	04A8	2320		COC	@MARGLM, 12	
	04AA	012E'				
0611	04AC	1601		JNE	#+4	
0612	04AE	058C		INC	12	

LINE SQUEEZER 948927-9901**

0613	04B0	C100		MOV	12, TMLOC	
0614	04B2	810C		C	12, LINPTR	ALL SQUEEZED UP?
0615	04B4	1405		JHE	SQ7	YES, PREPARE TO EXIT.
0616	04B6	C257		MOV	*TMLOC, 9	NO
0617	04B8	0289		CI	9, >FFFF	ANOTHER INACTIVE BLOCK
	04BA	FFFF				
0618	04BC	13AF		JEQ	SQ2	YES
0619	04BE	10BE		JMP	SQ4	NO
0620	04C0	C10A	SQ7	MOV	10, LINPTR	FIX LINPTR
0621	04C2	C260		MOV	@HICOR, 9	
	04C4	0008				
0622	04C6	6244		S	LINPTR, 9	
0623	04C8	C30B		MOV	RTN, 12	
0624	04CA	0289		CI	9, 80	ENOUGH SQUEEZED?
	04CC	0050				
0625	04CE	1B09		JH	SQ7A	
0626	04D0	06A0		BL	@MSGOUT	
	04D2	02CE				
0627	04D4	0012		DATA	18	BUFFER FULL
0628	04D6	C260		MOV	@FFLG, 9	CALLED BY FIND?
	04D8	0128				
0629	04DA	1602		JNE	#+6	YES
0630	04DC	0460		B	@COMMND	NO
	04DE	0226				
0631	04E0	050C		INCT	12	
0632	04E2	045C	SQ7A	B	*12	

```

0635      * TITLE:      FLIN
0636      *              FIND LINE ADDRESS
0637      * REVISION:
0638      *              ORIGINAL
0639      * COMPUTER:  990, ASSEMBLY
0640      * ABSTRACT:  FINDS LINE ADDRESS OF LINE WHOSE NUMBER
0641      *              IS IN NUMB. RESULTS RETURNED IN TMLOC.
0642      * CALLING SEQUENCE:
0643      *              @NUMB = LINE NUMBER
0644      *              BL  @FLIN
0645      04E4  C2A0  FLIN  MOV  @ANCHOR,10
           04E6  017E
0646      04E8  C1CA  FLIN1 MOV  10, TMLOC
0647      04EA  C297          MOV  *TMLOC,10      LOOK AT LINE NUMBER
0648      04EC  880A          C    10,@NUMB
           04EE  013C
0649      04F0  1306          JEQ  FDXT          THIS IS IT
0650      04F2  0227          RI  TMLOC,4          STEP POINTER AND
           04F4  0004
0651      04F6  C297          MOV  *TMLOC,10      LOOK AT NEXT LINE ADDR
0652      04F8  16F7          JNE  FLIN1          NOT LAST ITEM, LOOK SOME MORE
0653      04FA  0460          B    @NOTFND          LAST ITEM
           04FC  03F4
0654      04FE  045B  FDXT  B    *RTN          GO BACK
  
```

```

0657                    * TITLE:      SKANOP/SKANTM
0658                    *              SCAN OPERANDS
0659                    * REVISION:
0660                    *              ORIGINAL
0661                    * COMPUTER: 990, ASSEMBLY
0662                    * ABSTRACT: PARSSES SIMPLE OPERANDS WHICH CONSIST OF
0663                    *              A NUMBER FOLLOWED BY A CARRIAGE RETURN
0664                    * CALLING SEQUENCE:
0665                    *              BL    @SKANOP
0666                    *              BL    @SKANTM
0667    0500    C24B    SKANOP MOV    RTN, 9
0668    0502    06A0                    BL    @NUMEXT
                  0504    0526 ✓
0669    0506    C2A0                    MOV   @NUMBF, 10            NUMBER FOUND?
                  0508    013A ✓
0670    050A    1605                    JNE   NMEXIT            YES
0671    050C    028C                    CI    12, >0D            NO
                  050E    000D
0672    0510    1303                    JEQ   EXTSK            IF C/R, EXIT
0673    0512    0460                    B    @OPNERR            IF NOT, ERROR
                  0514    03E0 ✓
0674    0516    05C9    NMEXIT INCT 9            NUMBER FOUND EXIT
0675    0518    0459    EXTSK   B    *9            DEFAULT EXIT
0676    051A    C1CB    SKANTM MOV    RTN, TMLOC        SAVE RETURN
0677    051C    06A0                    BL    @SKANOP
                  051E    0500 ✓
0678    0520    0457                    B    *TMLOC            C/R FOUND
0679    0522    0460                    B    @OPNERR            NUMBER FOUND
                  0524    03E0 ✓
  
```

```

0682 * TITLE: NUMEXT
0683 * NUMBER EXTRACT
0684 * REVISION:
0685 * ORIGINAL
0686 * COMPUTER: 990, ASSEMBLY
0687 * ABSTRACT: EXTRACTS NUMBER FROM COMMAND INPUT
0688 * BUFFER AND CONVERTS IT TO BINARY
0689 * CALLING SEQUENCE:
0690 * BL @NUMEXT
0691 * STATISTICS: UPON COMPLETION WORKSPACE REGISTER 12
0692 * CONTAINS THE TERMINATING CHARACTER.
0693 * UDCNT CONTAINS OPERAND, AND NUMBF CONTAINS
0694 * NUMBER OF CHARACTERS. REGISTERS 9 AND
0695 * TMLOC MUST BE KEPT SAFE.
0696 0526 04C1 NUMEXT CLR UDCNT CLEAR NUMBER
0697 0528 04E0 CLR @NUMBF CLEAR NUMBER FOUND FLAG
052A 013A'
0698 052C C28B MOV RTN,10 SAVE RETURN
0699 052E 06A0 GETCH BL @GCHA GET A CHAR FROM COMBUF
0530 0580'
0700 0532 028C CI 12,>20 IS IT A BLANK?
0534 0020
0701 0536 13FB JEQ GETCH IF YES, TRY AGAIN
0702 0538 028C NCH CI 12,'0'
053A 0030
0703 053C 1110 JLT NXIT NOT NUMERIC
0704 053E 028C CI 12,'9'
0540 0039
0705 0542 1500 JGT NXIT NOT NUMERIC
0706 0544 05E0 INCT @NUMBF SET NUMBER FOUND
0546 013A'
0707 0548 3860 MPY @TEN,UDCNT CONVERT NUMBER
054A 017C'
0708 054C 024C ANDI 12,>000F GET RID OF LEADING ASCII BITS
054E 000F
0709 0550 A302 A LINAD,12 TO BINARY
0710 0552 C04C MOV 12,UDCNT
0711 0554 06A0 BL @GCHA GET NEXT CHAR
0556 0580'
0712 0558 10EF JMP NCH GO PROCESS IT
0713 055A 06A0 GETCH1 BL @GCHA
055C 0580'
0714 055E 028C NXIT CI 12,>20 FIND NON-BLANK
0560 0020
0715 0562 13FB JEQ GETCH1 TERMINATOR
0716 0564 028C CI 12,'0' DIGIT NOT VALID
0566 0030
0717 0568 1105 JLT OKXT TERMINATOR
0718 056A 028C CI 12,'9'
056C 0039
0719 056E 1502 JGT OKXT
0720 0570 0460 B @OPNERR
0572 03E0'
0721 0574 8801 OKXT C UDCNT,@LLIMIT LINE NUMBERS MUST
    
```

0722	0576	0182'			
0723	0578	1202	JLE	EXTEXT	NOT EXCEED 9999
0723	057A	0460	B	@OPNERR	
	057C	03E0'			
0724	057E	045A	EXTEXT B	*10	
0725			* EXTRACT CHARACTER		
0726	0580	C320	GCHA	MOV @CC,12	SETUP INDEX
	0582	0138'			
0727	0584	D32C	MOVB	@CPBUF(12),12	GET CHARACTER
	0586	006E'			
0728	0588	05A0	INC	@CC	INCREMENT INDEX
	058A	0138'			
0729	058C	098C	SRL	12,8	POSITION CHAR FOR COMPARES
0730	058E	045B	B	*RTN	

```

0733      * TITLE:      EXOP
0734      *              EXTRACT OPERAND
0735      * REVISION:
0736      *              ORIGINAL
0737      * COMPUTER:  990, ASSEMBLY
0738      * ABSTRACT:  ROUTINE TO EXTRACT LINE NUMBERS FOR THOSE
0739      *              COMMANDS WHICH EXPECT OPERANDS OF THE TYPES
0740      *              L-K (ABSOLUTE LINE NOS)
0741      *              +N (RELATIVE TO POINTER)
0742      *              -N (RELATIVE TO POINTER)
0743      *              N  (SAME AS +N)
0744      *              (POINTER LINE)
0745      *              ERROR EXIT TAKEN IF KCL, FIRST CHAR NOT
0746      *              NUMERIC, +, -, OR IF OPERAND SEPARATOR
0747      *              NOT - OR CR.
0748      * CALLING SEQUENCE:
0749      *              BL @EXOP
0750  0590  C14B  EXOP  MOV  RTN, 5          SAVE RETURN
0751  0592  C1D0          MOV  *DUMNXT, TMLOC  MEMORY EMPTY?
0752  0594  1602          JNE  $+6          NO
0753  0596  0460          B    @MEMPTY        YES
0754      0598  03EA'
0754  059A  04E0          CLR  @ABRLFG        SET ABSOLUTE-RELATIVE FLAG T
0755      059C  011C'
0755  059E  06A0          BL   @NUMEXT        GET FIRST DEPRAND
0756      05A0  0526'
0756  05A2  C2A0          MOV  @NUMBF, 10       NUMBER FOUND?
0757      05A4  013A'
0757  05A6  1635          JNE  ABNMS          YES
0758  05A8  028C          CI   12, >0D        NO
0759      05AA  0000
0759  05AC  132E          JEQ  CURPTL        IF C/R USE CURRENT POINTER VA
0760  05AE  028C          CI   12, '+'        EXPECT RELATIVE NUMBER
0761      05B0  002B
0761  05B2  1305          JEQ  RELNUM        IF FIRST CHAR IS + OR -,
0762  05B4  028C          CI   12, '-'        OTHERWISE, ERROR
0763      05B6  002D
0763  05B8  1302          JEQ  RELNUM
0764  05BA  0460          B    @OPNERR
0765      05BC  03E0'
0765  05BE  092C  RELNUM SRL  12, 2          CHANGE + - TO EVEN-ODD
0766  05C0  C24C          MOV  12, 9          AND SAVE
0767  05C2  06A0          BL   @NUMEXT        GET NUMBER
0768      05C4  0526'
0768  05C6  C2A0          MOV  @NUMBF, 10
0769      05C8  013A'
0769  05CA  1602          JNE  $+6
0770  05CC  0460          B    @OPNERR
0771      05CE  03E0'
0771  05D0  C801  NULREL MOV  UDCNT, @ABRLFG  SET FLAG TO RELATIVE
0772      05D2  011C'
0772  05D4  1602          JNE  $+6
0773  05D6  0460          B    @OPNERR        ZERO NOT ALLOWED
0773      05D8  03E0'
  
```



```

0774 05DA C006      MOV  CLLOC, PLIMTS SET LOCATION OF FIRST LINE
0775 05DC 0501      NEG  UDCNT
0776 05DE 2260      COC  @MARGLM, 9
      05E0 012E
0777 05E2 1608      JNE  EXOPEX
0778                      * START MOVING UPLINKED LIST,
0779                      * TO FIND ADDR OF FIRST LINE
0780 05E4 C1C3      RELNEG MOV  PLIMTS, TMLOC
0781 05E6 05C7      RLNEG1 INCT TMLOC
0782 05E8 C1D7      MOV  *TMLOC, TMLOC GET ADDR OF LINE
0783 05EA C297      MOV  *TMLOC, 10      DUMMY?
0784 05EC 1306      JEQ  EX1              YES
0785 05EE 0581      INC  UDCNT
0786 05F0 16FA      JNE  RLNEG1
0787 05F2 C0C7      MOV  TMLOC, PLIMTS
0788 05F4 C060      EXOPEX MOV  @ABRLFG, UDCNT SET PLIMTS EQUAL
      05F6 011C
0789 05F8 1020      JMP  EXEXOP          EXIT
0790                      *
0791 05FA C0D0      EX1  MOV  *DUMNXT, PLIMTS SET PLIMTS TO TOP LIMITS
0792 05FC 06A0      BL   @ATTOP          PRINT 'OFF THE TOP'
      05FE 03D6
0793 0600 A060      A    @ABRLFG, UDCNT  CALC NO OF LINES INVOLVED
      0602 011C
0794 0604 161A      JNE  EXEXOP
0795 0606 0460      B    @COMMND        NO LINES TO WORK WITH
      0608 0226
0796 060A C0C6      CURPTL MOV  CLLOC, PLIMTS USE CURRENT LINE AS OPERAND
0797 060C 05A0      INC  @ABRLFG        SET RELATIVE FLAG
      060E 011C
0798 0610 10F1      JMP  EXOPEX
0799 0612 C0C1      ABNMS MOV  UDCNT, PLIMTS SAVE LOWER LIMIT
0800 0614 028C      CI   12, '-'
      0616 002D
0801 0618 1302      JEQ  GULM           GET UPPER LIMIT
0802 061A 04C9      CLR  9
0803 061C 10D9      JMP  NULREL        NULL-SIGNED RELATIVE LINE NUMB
0804 061E C0C3      GULM MOV  PLIMTS, PLIMTS TRYING TO OPERATE ON LINE ZERO?
0805 0620 130D      JEQ  EREXOP        NO
0806 0622 06A0      BL   @NUMEXT       GET SECOND OPERAND
      0624 0526
0807 0626 C2A0      MOV  @NUMBF, 10
      0628 013A
0808 062A 1308      JEQ  EREXOP        ERROR
0809 062C 80C1      C    UDCNT, PLIMTS
0810 062E 1106      JLT  EREXOP
0811 0630 C803      MOV  PLIMTS, @NUMB
      0632 013C
0812 0634 06A0      BL   @FLIN        FIND LINE ADDR
      0636 04E4
0813 0638 C0C7      MOV  TMLOC, PLIMTS
0814 063A 0455      EXEXOP B *5
0815 063C 0460      EREXOP B @OPNERR
      063E 03E0

```

```

0818          * TITLE:      BTM
0819          *              BOTTOM COMMAND
0820          * REVISION:
0821          *              ORIGINAL
0822          * COMPUTER:  990, ASSEMBLY
0823          * ABSTRACT:  MOVES POINTER TO LAST ACTIVE LINE
0824          *              IN SOURCE LINE BUFFER
0825          * CALLING SEQUENCE:
0826          *              BLWP @BTM
0827 0640 0026' BTM      DATA ROUTRG, $+2
          0642 0644'
0828 0644 06A0          BL      @SKANTM          CHECK TERM CHAR
          0646 051A'
0829 0648 C1D0          MOV     *DUMNXT, TMLOC      MEMORY EMPTY?
0830 064A 1602          JNE     $+6                NO
0831 064C 0460          B       @MEMPTY           YES
          064E 03EA'
0832 0650 06A0          BL      @BOTTOM           FIND BOTTOM LINE
          0652 065A'
0833 0654 C217          MOV     *TMLOC, CLINNO     SET POINTERS
0834 0656 C187          MOV     TMLOC, CLLOC
0835 0658 0380          RTWP
0836          *
0837          * ROUTINE THAT FINDS BOTTOM LINE
0838          *
0839 065A C2A0          BOTTOM MOV @ANCHOR, 10        GET ADDR OF FIRST LINE
          065C 017E'
0840 065E 022A          BTM1   AI      10, 4          POINTS TO LOC THAT CONTAINS ADDR
          0660 0004
0841 0662 C1CA          MOV     10, TMLOC      SAVE IT
0842 0664 C29A          MOV     *10, 10        HAS NEXT LINE BEEN BROUGHT IN?
0843 0666 16FB          JNE     BTM1         YES MOVE DOWN ANOTHER ONE
0844 0668 0227          AI      TMLOC, -4      SET POINTER TO BEGINNING OF LINE
          066A FFFC
0845 066C 045B          B       *RTN          RETURN
  
```

CHANGE COMMAND

948927-9901**

```
0848          * TITLE:      CHNG
0849          *             CHANGE COMMAND
0850          * REVISION:
0851          *             ORIGINAL
0852          * COMPUTER:  990, ASSEMBLY
0853          * ABSTRACT: CHANGES SOURCE LINES VIA REMOVE
0854          *             AND INSERT
0855          * CALLING SEQUENCE:
0856          *             BLWP @CHNG
0857 066E 0026' CHNG  DATA ROUTRG, #+2
          0670 0672'
0858 0672 05A0          INC  @CHFLG          SET CHANGE FLAG
          0674 0116'
0859 0676 06A0          BL   @RMVCH          DO REMOVES
          0678 0E38'
0860 067A 06A0          BL   @INSCH          DO INSRTS
          067C 0A80'
0861 067E 0380          RTWP
```

```

0864 * TITLE: DWN
0865 * DOWN COMMAND
0866 * REVISION:
0867 * ORIGINAL
0868 * COMPUTER: 990, ASSEMBLY
0869 * ABSTRACT: MOVES POINTER DOWN NUMBER OF LINES
0870 * SPECIFIED IN COMMAND TO MOVE PAST
0871 * LAST LINE, REQUIRED NUMBER OF SOURCE
0872 * LINES ARE BROUGHT IN
0873 * CALLING SEQUENCE:
0874 * BLWP @DWN
0875 0680 0026' DWN DATA ROUTRG, $+2
0682 0684'
0876 0684 06A0 BL @UDCOM1 DETERMINE COUNT
0686 06BC'
0877 0688 C286 MOV CLLOC, 10 SET UP SEARCH
0878 068A 022A DWN1 AI 10, 4 POINTERS
068C 0004
0879 068E C1CA MOV 10, TMLOC
0880 0690 C297 MOV *TMLOC, 10
0881 0692 0601 DEC UDCNT FINISHED?
0882 0694 1509 JGT DWN2 NO MOVE DOWN ONE MORE
0883 0696 C28A MOV 10, 10 IS NEXT LINE IN?
0884 0698 160A JNE DWN4
0885 069A 0581 DWN3 INC UDCNT NO
0886 069C A060 A @MAXLIN, UDCNT LINES TO BRING IN
069E 015C'
0887 06A0 06A0 BL @INSOU BRING THEM IN
06A2 0708'
0888 06A4 1007 JMP EOFEXT END OF FILE EXIT
0889 06A6 1009 JMP EXIT
0890 06A8 C28A DWN2 MOV 10, 10
0891 06AA 13F7 JEQ DWN3
0892 06AC 10EE JMP DWN1
0893 06AE C18A DWN4 MOV 10, CLLOC SET POINTERS
0894 06B0 C216 MOV *CLLOC, CLINNO
0895 06B2 1003 JMP EXIT
0896 06B4 06A0 EOFEXT BL @MSGOUT
06B6 02CE'
0897 06B8 000C DATA 12
0898 06BA 0380 EXIT RTWP
0899 * ROUTINE COMMON TO UP AND DOWN
0900 06BC C14B UDCOM1 MOV RTN, 5
0901 06BE 06A0 BL @SKANOP
06C0 0500'
0902 06C2 0581 INC UDCNT DEFAULT COUNT
0903 06C4 C041 MOV UDCNT, UDCNT
0904 06C6 13F9 JEQ EXIT
0905 06C8 0455 B *5

```

```

0908          * TITLE:      UP
0909          *              UP COMMAND
0910          * REVISION:
0911          *              ORIGINAL
0912          * COMPUTER: 990, ASSEMBLY
0913          * ABSTRACT: MOVES POINTER UP NUMBER OF LINES SPECIFIED
0914          *              BY COMMAND
0915          * CALLING SEQUENCE:
0916          *              BLWP @UP
0917 06CA 0026' UP      DATA ROUTRG, $+2
          06CC 06CE'
0918 06CE 06A0          BL      @UDCOM1          DETERMINE COUNT
          06D0 06BC'
0919 06D2 C1D0          MOV     *DUMNXT, TMLOC BUFFER EMPTY?
0920 06D4 1602          JNE     $+6
0921 06D6 0460          B       @MEMPTY
          06D8 03EA'
0922 06DA 8187          C       TMLOC, CLLOC  ALREADY AT TOP?
0923 06DC 1603          JNE     UP4
0924 06DE 06A0          BL      @ATTOP          YES
          06E0 03D6'
0925 06E2 100B          JMP     EXTUP
0926 06E4 C1C6  UP4     MOV     CLLOC, TMLOC  GET LOC OF LINE
0927 06E6 05C7          INCT  TMLOC          AT POINTER
0928 06E8 C297          MOV     *TMLOC, 10  SET TO LOOK AT PREV LINE
0929 06EA 05CA  UP1     INCT  10
0930 06EC C1CA          MOV     10, TMLOC
0931 06EE C297          MOV     *TMLOC, 10
0932 06F0 0601          DEC     UDCNT
0933 06F2 1504          JGT     UP2          MOVE UP ONE MORE
0934 06F4 0647          DECT  TMLOC        SET POINTERS.
0935          UP3
0936 06F6 C187          MOV     TMLOC, CLLOC
0937 06F8 C216          MOV     *CLLOC, CLINNO
0938 06FA 0380  EXTUP   RTWP
0939 06FC C69A  UP2     MOV     *10, *10          TERMINAL NODE?
0940 06FE 16F5          JNE     UP1          NO
0941 0700 06A0          BL      @ATTOP          YES
          0702 03D6'
0942 0704 C1D0          MOV     *DUMNXT, TMLOC  SET TO TOP LINE.
0943 0706 10F7          JMP     UP3

```

```

0946          * TITLE:      INSOU
0947          *              INPUT SOURCE LINE
0948          * REVISION:
0949          *              ORIGINAL
0950          *              03/15/76
0951          *              MODIFIED TO RUN WITH PXRMT
0952          * COMPUTER: 990, ASSEMBLY
0953          * ABSTRACT: BRINGS IN SOURCE LINES, SETS UP HEADERS,
0954          *              SETS UP POINTERS
0955          * CALLING SEQUENCE:
0956          *              BL @INSOU
0957 0708 C14B  INSOU  MOV  RTN,5          SAVE RETURN ADDRESS
0958 070A C820          MOV  @EOFLG,@EOFLG      CHECK EOF FLAG
          070C 0114 ✓
          070E 0114 ✓
0959 0710 1640          JNE  EXITIN          IF SET, EXIT
0960 0712 06A0          BL   @BOTTOM        FIND LAST LINE
          0714 065A ✓
0961 0716 C807          MOV  TMLOC,@PRVLIN
          0718 015E ✓
0962 071A 0227          RI   TMLOC,4
          071C 0004
0963 071E C807          MOV  TMLOC,@SLOC
          0720 0156 ✓
0964 0722 C2A0  INPS3  MOV  @HICOR,10      IS THERE ROOM
          0724 0008 ✓
0965 0726 6284          S    LINPTR,10      FOR ONE LINE?
0966 0728 028A          CI   10,80          MIN SPACE FOR ONE LINE
          072A 0050
0967 072C 1B02          JH   #+6
0968 072E 06A0          BL   @SQUEEZ        NO
          0730 0408 ✓
0969 0732 05A0          INC  @MAXLIN
          0734 015C ✓
0970 0736 C284          MOV  LINPTR,10
0971 0738 CE90          MOV  @MAXLIN,*10+    PUT LINE NO IN HEADER
          073A 015C ✓
0972 073C CE90          MOV  @PRVLIN,*10+    SET PREVIOUS LINE PTR
          073E 015E ✓
0973 0740 C80A          MOV  10,@SLINPT     SAVE LOC OF CELL WHICH MAY REQ
          0742 0158 ✓
0974 0744 04FA          CLR  *R10+          ADDR FOR INPUT IN R10
0975 0746 C30A          MOV  R10,R12        SAVE INPUT ADDR
0976 0748 0420          BLWP @READ          INPUT
          074A 0F9A ✓
0977 074C 0249          ANDI R9,EOF+IOERR   CHECK FOR EOF OR IO ERR
          074E 6000
0978 0750 151E          JGT  INPS1          YES
0979 0752 060A  INPS2  DEC  R10          DELETE TRAILING BLANKS
0980 0754 981A          CB   *R10,@SP
          0756 0186 ✓
0981 0758 13FC          JEQ  INPS2
0982 075A 058A          INC  R10
0983 075C DE90          MOVB @CR,*R10+     END SOURCE LINE WITH CR

```

0984	075E	0185				
	0760	22A0	COC	@MARGLM, R10	INC TO	
	0762	012E				
0985	0764	1601	JNE	#+4	NEXT WORD	
0986	0766	058A	INC	R10	IF ODD ADDRESS	
0987	0768	C804	MOV	LINPTR, @PRVLIN		
	076A	015E				
0988	076C	C10A	MOV	R10, LINPTR	POINTER FOR NEXT SOURCE LINE	
0989	076E	C2A0	MOV	@SLOC, R10		
	0770	0156				
0990	0772	C6A0	MOV	@PRVLIN, *R10	FIX NXT LINE PTR OF PRV LINE	
	0774	015E				
0991	0776	C820	MOV	@SLINPT, @SLOC		
	0778	0158				
	077A	0156				
0992	077C	C220	MOV	@MAXLIN, CLINNO	SET POINTER	
	077E	015C				
0993	0780	C1A0	MOV	@PRVLIN, CLLOC	PARAMETERS	
	0782	015E				
0994	0784	8801	C	UDCNT, @MAXLIN	NEED ANOTHER LINE?	
	0786	015C				
0995	0788	150C	JGT	INPS3	YES	
0996	078A	0505	INCT	5		
0997	078C	0455	B	*5	RETURN	
0998	078E	05A0	INPS1	INC	@EOFLG	EOF RECORD DETECTED
	0790	0114				
0999	0792	0455	EXITIN	B	*5	

```

1002      * TITLE:      EXT
1003      *              EXIT COMMAND
1004      * REVISION: 03/15/76
1005      *              ORIGINAL - COMMAND ADDED TO RESTART WITHOUT
1006      *              WRITING EOF
1007      * COMPUTER: 990, 990 ASSEMBLY
1008      * ABSTRACT:  RESTARTS THE EDIT PROCESS WITHOUT WRITING EOF
1009      * CALLING SEQUENCE:
1010      *              BLWP @EXT
1011      0794 0006' EXT      DATA MAINRG.ENDACT      WORKSPACE, START
1011      0796 027A'
1012      *

```



```

1015 * TITLE: FND
1016 * FIND COMMAND
1017 * REVISION:
1018 * ORIGINAL
1019 * 03/15/76
1020 * MODIFIED TO RUN WITH PXRMTN
1021 * COMPUTER: 990, ASSEMBLY
1022 * ABSTRACT: THE FIND COMMAND LOCATES STRINGS AND
1023 * THEN CAN REPLACE THE FIRST STRING BY
1024 * THE SECOND WITH VERIFY AND PRINT OPTIONS.
1025 * CALLING SEQUENCE:
1026 * BLWP @FND
1027 FND
1028 0798 0020' DATA R0UTR1, $+2
      079A 0790'
1029 079C 02E0 LWPI R0UTRG
      079E 0026'
1030 07A0 06A0 BL @EXOP GET OPERANDS
      07A2 0590'
1031 07A4 C808 MOV CLINNO, @R0UTRG-2
      07A6 0024'
1032 07A8 028C CI 12, 'L' FIND IN LABEL FIELD?
      07AA 004C
1033 07AC 1603 JNE $+8
1034 07AE 0720 SETO @LFFLG YES
      07B0 011E'
1035 07B2 1007 JMP FNDA
1036 07B4 028C CI 12, 'F' FIND FIRST OCCURANCE?
      07B6 0046
1037 07B8 1302 JEQ $+6
1038 07BA 0460 B @EREXFN NO
      07BC 0A58'
1039 07BE 04E0 CLR @LFFLG YES
      07C0 011E'
1040 07C2 04E0 FNDA CLR @SRKNT CLEAR STRINGS-FOUND COUNT
      07C4 0166'
1041 07C6 04E0 CLR @PFLG
      07C8 0120'
1042 07CA 04E0 CLR @VFLG
      07CC 0122'
1043 07CE 04E0 CLR @REPFLG CLEAR REPLACEMENT FLAG
      07D0 0124'
1044 07D2 04E0 CLR @NWLFL CLEAR NEW LOC FLAG
      07D4 0126'
1045 07D6 06A0 BL @GITCR GET AND SAVE
      07D8 0A5C'
1046 07DA C80C MOV 12, @FSD FIRST STRING DELIMETER
      07DC 0164'
1047 07DE C820 MOV @CC, @FSDX
      07E0 0138'
      07E2 0174'
1048 07E4 06A0 FNDB BL @GITCR SCAN TO
      07E6 0A6A'
1049 07E8 880C C 12, @FSD END OF

```

1050	07EA	0164'			
	07EC	16FB	JNE	FNDB	FIRST STRING
1051	07EE	C2A0	MOV	@CC, 10	CHECK FOR NULL STRING
	07F0	0138'			
1052	07F2	060A	DEC	10	ADJUST FOR DELIMETER
1053	07F4	62A0	S	@FSDX, 10	
	07F6	0174'			
1054	07F8	1602	JNE	#+6	
1055	07FA	0460	B	@OPNERR	NOT ALLOWED
	07FC	03E0'			
1056	07FE	C1CA	MOV	10, TMLOC	SAVE STRING LENGTH.
1057					
			FNDC		
1058	0800	06A0	BL	@GCHA	GET NEXT CHAR.
	0802	0580'			
1059	0804	028C	CI	12, >20	CHAR FROM COMBUF
	0806	0020			
1060	0808	13FB	JEQ	FNDC	
1061	080A	028C	CI	12, 'V'	'V' IS NOT A VALID
	080C	0056			
1062	080E	1602	JNE	#+6	DELIMETER FOR
1063	0810	0460	B	@OPNERR	SECOND STRING
	0812	03E0'			
1064	0814	028C	CI	12, >0D	IF C/R, THEN NO PRINT, NO VERI
	0816	000D			
1065	0818	132H	JEQ	FND1B	AND NO STRING REPLACEMENT
1066	081H	028C	CI	12, 'P'	CHECK FOR PRINT OPTION.
	081C	0050			
1067	081E	1320	JEQ	FND1B-4	
1068	0820	C80C	MOV	12, @SSD	SAVE 2ND
	0822	0168'			
1069	0824	C820	MOV	@CC, @SSDX	AND ITS POSITION.
	0826	0138'			
	0828	0176'			
1070	082A	06A0	FNDF	@GITCRB	SCAN TO END OF
	082C	0A6A'			
1071	082E	880C	C	12, @SSD	SECOND STRING.
	0830	0168'			
1072	0832	16FB	JNE	FNDF	
1073	0834	05A0	INC	@REPFLG	SET STRING REPLACEMENT FLAG
	0836	0124'			
1074	0838	C2A0	MOV	@CC, 10	
	083A	0138'			
1075	083C	060A	DEC	10	ADJUST FOR DELIMETER.
1076	083E	62A0	S	@SSDX, 10	STRINGS WITH
	0840	0176'			
1077	0842	81CA	C	10, TMLOC	SAME LENGTH?
1078	0844	1302	JEQ	#+6	
1079	0846	05A0	INC	@NWLF	SET NEW-LOC FLAG
	0848	0126'			
1080					
			FASK		
1081	084A	06A0	BL	@GCHA	SEE IF VERIFY OR
	084C	0580'			
1082	084E	028C	CI	12, >0D	PRINT OPTION SET.
	0850	000D			
1083	0852	130D	JEQ	FND1B	NEITHER

```
1084 0854 028C          CI  12, 'V'
      0856 0056
1085 0858 1602          JNE  $+6
1086 085A 05A0          INC  @VFLG          SET VERIFY
      085C 0122'
1087 085E 028C          CI  12, 'P'
      0860 0050
1088 0862 1602          JNE  $+6
1089 0864 05A0          INC  @PFLG          SET PRINT
      0866 0120'
1090 0868 10F0          JMP  FASK
1091                                * LOCATE THE LINE TO BE SEARCHED
1092 086A 05A0          INC  @PFLG          SET PRINT
      086C 0120'
1093 086E 05A0 FND1B  INC  @FFLG
      0870 0128'
1094 0872 C243          MOV  PLIMTS, 9
1095 0874 C089 FND1   MOV  9, LINAD          SAVE ADDR OF CURRENT LINE
1096 0876 C260          MOV  @NWL, 9          EXTRA MEMORY SPACE REQUIRED?
      0878 0126'
1097 087A 130F          JEQ  FND1A          NO
1098 087C C2A0          MOV  @HICOR, 10      ROOM FOR
      087E 0008'
1099 0880 C120          MOV  @VLNPTR, LINPTR ONE
      0882 002E'
1100 0884 6284          S    LINPTR, 10      LINE?
1101 0886 028A          CI  10, 80
      0888 0050
1102 088A 1B07          JH   FND1A
1103 088C C220          MOV  @ROUTRG-2, CLINNO GET BACK LINE NO.
      088E 0024'
1104 0890 06A0          BL   @SQUEEZ          NO
      0892 0408'
1105 0894 1002          JMP  $+6
1106 0896 0460          B    @PFKNT          BUFFER FULL
      0898 0A40'
1107 089A C282 FND1A  MOV  LINAD, 10
1108 089C C83A          MOV  *10+, @CURNO    SAVE LINE NUMBER
      089E 0150'
1109 08A0 C80A          MOV  10, @PRLAD
      08A2 0152'
1110 08A4 C83A          MOV  *10+, @PRELAD   SAVE ADDR OF PREVIOUS LINE
      08A6 0148'
1111 08A8 C80A          MOV  10, @FXLAD
      08AA 016A'
1112 08AC C83A          MOV  *10+, @NXTLAD   SAVE ADDR OF NEXT LINE
      08AE 014E'
1113 08B0 C80A          MOV  10, @THLAD
      08B2 016C'
1114 08B4 04CD          CLR  FGPOS          CLEAR LINE
1115 08B6 04CE          CLR  FPPOS          INDICES
1116                                * CREATE HEADER FOR POSSIBLE NEW LINE
1117 08B8 C3C4          MOV  LINPTR, NEWPTR
1118 08BA C260          MOV  @NWL, 9
      08BC 0126'
```

1119	08BE	1601		JNE	#+4	
1120	08C0	C3C2		MOV	LINAD, NEWPTR	SPACE?
1121	08C2	CFE0		MOV	@CURNO, *NEWPTR+	COPY LINE NO TO HEADER
	08C4	0150				
1122	08C6	CFE0		MOV	@PRELAD, *NEWPTR+	FIX POINTER TO PREV LINE
	08C8	0148				
1123	08CA	CFE0		MOV	@NXTLAD, *NEWPTR+	FIX POINTER TO NEXT LINE
	08CC	014E				
1124	08CE	06A0	FND2	BL	@FROLIN	GET CHAR FROM OLD LINE
	08D0	0994				
1125	08D2	880D		C	FGPOS, @LOMRG	TIME TO START STRING MATCH
	08D4	0134				
1126	08D6	1406		JHE	FND3	YES
1127	08D8	06A0	FND4	BL	@TONLIN	NO
	08DA	09A8				
1128	08DC	0288		CI	FCHAR, >0D	
	08DE	0000				
1129	08E0	134B		JEQ	FND9	END OF OLD LINE
1130	08E2	10F5		JMP	FND2	
1131	08E4	880D	FND3	C	FGPOS, @HIMRG	TIME TO STOP STRING MATCHI
	08E6	0136				
1132	08E8	1547		JGT	FND9	YES, STRING NOT FOUND
1133	08EA	C820		MOV	@FSDX, @FSDXT	NO
	08EC	0174				
	08EE	0172				
1134	08F0	06A0		BL	@FRSTR	GET CHAR FROM STRING
	08F2	09B6				
1135	08F4	8308		C	FCHAR, 12	
1136	08F6	1304		JEQ	FND5	STRING MATCH FOUND
1137	08F8	C320	FND7	MOV	@LFFLG, 12	NO, IS MODE SET TO LABEL
	08FA	011E				
1138	08FC	113D		JLT	FND9	FIELD ONLY? YES NO MATCH
1139	08FE	10E0		JMP	FND4	TRY TO MATCH NEXT POSITIO
1140	0900	C80D	FND5	MOV	FGPOS, @SCFP	FOUND POINTER
	0902	016E				
1141	0904	0620		DEC	@SCFP	
	0906	016E				
1142	0908	06A0	FND6	BL	@TONLIN	
	090A	09A8				
1143	090C	C1CD		MOV	FGPOS, TMLOC	
1144	090E	06A0		BL	@FROLIN	
	0910	0994				
1145	0912	8807		C	TMLOC, @HIMRG	DON'T LOOK PAST HIGH MARGIN
	0914	0136				
1146	0916	1530		JGT	FND9	
1147	0918	06A0		BL	@FRSTR	
	091A	09B6				
1148	091C	8320		C	@FSD, 12	FOUND END OF FIRST STRING?
	091E	0164				
1149	0920	1309		JEQ	SFND	YES, MATCH IS FOUND.
1150	0922	8308		C	FCHAR, 12	
1151	0924	13F1		JEQ	FND6	MATCHING CONTINUES
1152	0926	C360		MOV	@SCFP, FGPOS	RESET OLD-LINE
	0928	016E				
1153	092A	058D		INC	FGPOS	SCAN

1154	092C	C380		MOV	FGPOS, FPOS	POSITION
1155	092E	06A0		BL	@FROLIN	GET CHAR
	0930	0994				
1156	0932	10D8		JMP	FND3	MATCHING STRING ENDED
1157	0934	05A0	SFND	INC	@SRKNT	STEP FOUND COUNT
	0936	0166				
1158	0938	C320		MOV	@VFLG, 12	
	093A	0122				
1159	093C	1300		JEQ	NOVY	VERIFY WAS NOT SET
1160	093E	C282		MOV	LINAD, 10	
1161	0940	06A0		BL	@PRLIN	PRINT OLD LINE
	0942	0C68				
1162	0944	020A		LI	R10, MSG13	PRINT 'Y/N?'
	0946	03B4				
1163	0948	0420		BLWP	@PRINTN	
	094A	0FF0				
1164	094C	06A0		BL	@GETANS	
	094E	02DA				
1165	0950	9820		CB	@PBUF, @YEA	DID HE ANSWER YES?
	0952	00C2				
	0954	0195				
1166	0956	1610		JNE	FND9	NO
1167	0958	C320	NOVY	MOV	@REPFLG, 12	REPLACE STRING?
	095A	0124				
1168	095C	1347		JEQ	FND11	NO
1169	095E	C820		MOV	@SSDX, @FSDXT	SET STRING EXTRACT POINTER
	0960	0176				
	0962	0172				
1170	0964	C3A0		MOV	@SCFP, FPOS	TO 2ND STRING.
	0966	016E				
1171	0968	06A0	FND8	BL	@FRSTR	GET CHAR FROM 2ND STRING
	096A	09B6				
1172	096C	8320		C	@SSD, 12	AT END?
	096E	0168				
1173	0970	132E		JEQ	FND10+4	YES
1174	0972	06A0		BL	@PTONLN	NO, COPY STRING 2 INTO
	0974	09A2				
1175	0976	10F8		JMP	FND8	NEW LINE, GET NEXT ONE.
1176	0978	C320	FND9	MOV	@ABRLFG, 12	ABSOLUTE OR RELATIVE?
	097A	011C				
1177	097C	1608		JNE	FND9A	RELATIVE
1178	097E	8060		C	@CURNO, UDCNT	ALL DONE?
	0980	0150				
1179	0982	135E		JEQ	PFKNT	PRINT FOUND-COUNT
1180	0984	C260	FND9B	MOV	@NXTLAD, 9	NO, PREPARE FOR NEXT LINE
	0986	014E				
1181	0988	1358		JEQ	FERXIT	THERE IS NO NEXT ONE
1182	098A	0460		B	@FND1	YES, DO NEXT ONE.
	098C	0874				
1183	098E	0601	FND9A	DEC	UDCNT	DECREMENT LINE COUNT.
1184	0990	15F9		JGT	FND9B	NOT FINISHED
1185	0992	1056		JMP	PFKNT	PRINT FOUND-COUNT
1186						
1187	0994	C2A0	* GET CHAR FROM OLD LINE	FROLIN MOV	@THLAD, 10	ADDR FOR LINE
	0996	016C				

1188	0998	A28D	A	FGPOS.10	ADD INDEX	
1189	099A	D21A	MOVB	*10,FCHAR	GET CHAR	
1190	099C	0888	SRA	FCHAR.8		
1191	099E	058D	INC	FGPOS	INCREMENT INDEX TO NEXT CHAR	
1192	09A0	045B	B	*RTN	RETURN	
1193			* PUT CHAR IN NEW LINE			
1194	09A2	C320	PTONLN	MOV	@STRCHR.12	
	09A4	0170				
1195	09A6	1001	JMP	\$\$+4		
1196	09A8	C308	TONLIN	MOV	FCHAR.12	
1197	09AA	C28F	MOV	NEWPTR.10	ADDR OF LINE	
1198	09AC	A28E	A	FPPOS.10	ADD INDEX	
1199	09AE	0A8C	SLA	12.8	POSITION CHAR	
1200	09B0	D68C	MOVB	12,*10	PUT IN LINE	
1201	09B2	058E	INC	FPPOS	INCREMENT INDEX	
1202	09B4	045B	B	*RTN		
1203			* GET A CHAR FROM STRING IN COMBUF			
1204			FRSTR			
1205	09B6	C2A0	MOV	@FSDXT.10		
	09B8	0172				
1206	09BA	D32A	MOVB	@CPBUF(10).12	GET CHAR	
	09BC	006E				
1207	09BE	088C	SRA	12.8		
1208	09C0	C80C	MOV	12,@STRCHR	SAVE	
	09C2	0170				
1209	09C4	05A0	INC	@FSDXT	INCREMENT INDEX	
	09C6	0172				
1210	09C8	045B	B	*RTN	RETURN	
1211	09CA	06A0	FND10	BL	@FROLIN	COPY REMAINDER OF OLD
	09CC	0994				
1212	09CE	06A0	BL	@TONLIN	LINE ONTO NEW LINE	
	09D0	09A8				
1213	09D2	0288	CI	FCHAR.>0D		
	09D4	000D				
1214	09D6	16F9	JNE	FND10		
1215	09D8	028E	CI	FPPOS.81		
	09DA	0051				
1216	09DC	1107	JLT	FND11	LINE WILL FIT	
1217	09DE	0208	LI	FCHAR.>0D	LINE TOO LONG	
	09E0	000D				
1218	09E2	020E	LI	FPPOS.80		
	09E4	0050				
1219	09E6	06A0	BL	@TONLIN	AFTER 80TH CHAR	
	09E8	09A8				
1220	09EA	1003	JMP	FND11B		
1221			FND11			
1222	09EC	C320	MOV	@PFLG.12		
	09EE	0120				
1223	09F0	1305	JEQ	FND11A	NO	
1224			FND11B			
1225	09F2	C28F	MOV	NEWPTR.10	YES	
1226	09F4	022A	AI	10,-6		
	09F6	FFFA				
1227	09F8	06A0	BL	@PRLIN		
	09FA	0C68				

```

1238 * LINK THE NEW LINE INTO LIST MEMORY WHERE THE
1239 * OLD ONE WAS, AND MARK OLD LINE INACTIVE
1230 09FC C320 FND11A MOV @REPFLG, 12 ANY REPLACEMENT?
      09FE 0124
1231 0A00 13BB JEQ FND9 NO
1232 0A02 C320 MOV @NWLF, 12 NEW LOCATION?
      0A04 0126
1233 0A06 13B8 JEQ FND9 NO
1234 0A08 C320 MOV @PRLAD, 12
      0A0A 0152
1235 0A0C C31C MOV *12, 12
1236 0A0E 022C AI 12, 4
      0A10 0004
1237 0A12 C704 MOV LINPTR, *12 FIX PREV-LIN
1238 0A14 C320 MOV @FXLAD, 12 FIX NEXT LINE
      0A16 016A
1239 0A18 C31C MOV *12, 12
1240 0A1A 1302 JEQ FND11C NO NEXT LINE
1241 0A1C 05CC INCT 12
1242 0A1E C704 MOV LINPTR, *12
1243 0A20 C142 FND11C MOV LINAD, 5
1244 0A22 C220 MOV @ROUTRG-2, CLINNO
      0A24 0024
1245 0A26 06A0 BL @KLRLIN CLEAR THE LINE
      0A28 0E16
1246 0A2A C30E MOV FPPOS, 12 ADJUST
1247 0A2C 2320 COC @MARGLM, 12 PUT ON WORD BOUNDRY
      0A2E 012E
1248 0A30 1601 JNE $+4
1249 0A32 058C INC 12
1250 0A34 A30F A NEWPTR, 12 POINTER
1251 0A36 C10C MOV 12, LINPTR
1252 0A38 109F JMP FND9 CHECK IF ALL DONE
1253 *
1254 0A3A 06A0 FERXIT BL @MSGOUT
      0A3C 02CE
1255 0A3E 0010 DATA 16 PRINT 'LAST LINE'
1256 0A40 C260 PFKNT MOV @SRKNT, 9
      0A42 0166
1257 0A44 06A0 BL @CONVRT CONVERT COUNT TO ASCII
      0A46 0004
1258 0A48 06A0 BL @MSGOUT PRINT MESSAGE
      0A4A 02CE
1259 0A4C 000E DATA 14 0000 FOUND
1260 0A4E C220 MOV @ROUTRG-2, CLINNO GET BACK LINE NO.
      0A50 0024
1261 0A52 02E0 LMPI ROUTR1
      0A54 002C
1262 0A56 0380 RTMP
1263 0A58 0460 EREXFN B @OPNERR
      0A5A 03E0
1264 0A5C C14B GITCR MOV RTN, 5
1265 0A5E 06A0 GITCR1 BL @GITCRB EXTRACT CHAR FROM
      0A60 0A6A
1266 0A62 028C CI 12, >20 COMBUF, REJECTING
  
```

	0A64	0020				
1267	0A66	13FB	JEQ	GITCR1		BLANKS
1268	0A68	0455	B	*5		
1269	0A6A	C24B	GITCRB	MOV	RTN, 9	
1270	0A6C	06A0	BL	@GCHA		EXTRACT CHAR.
	0A6E	0580				
1271	0A70	028C	CI	12.>0D		FROM COMBUF, INCLUDING BLANKS
	0A72	0000				
1272	0A74	1602	JNE	*+6		BUT NOT C/R
1273	0A76	0460	B	@OPNERR		
	0A78	03E0				
1274	0A7A	0459	B	*9		


```

1277 * TITLE:      INSRT
1278 *             INSERT COMMAND
1279 * REVISION:
1280 *             ORIGINAL
1281 *             03/15/76
1282 *             MODIFIED TO RUN WITH PXRMTX
1283 * COMPUTER:  990, ASSEMBLY
1284 * ABSTRACT:  INSERTS LINES FROM KEYBOARD INTO SOURCE
1285 *             LINE BUFFER AND SETS UP POINTERS
1286 * CALLING SEQUENCE:
1287 *             BLWP @INSRT
1288 *             BL  @INSCH
1289 * STATISTICS: LINE NUMBER OF INSERTED LINES = >8000
1290 0A7C 0026 ✓ INSRT DATA ROUTRG, INSR5
1291 0A7E 0A8A ✓
1291 0A80 C80B  INSCH MOV  RTN,@SAVRTN      ENTRY POINT WHEN CALLED BY CHAN
1292 0A82 0154 ✓
1292 0A84 C1E0      MOV  @CHFLG, TMLOC
1293 0A86 0116 ✓
1293 0A88 1008      JMP  INSR1
1294 0A8A C086  INSR5 MOV  CLLOC, LINAD  GET POINTER LINE ADDR
1295 0A8C 06A0      BL   @SKANOP           GET OPERAND
1296 0A8E 0500 ✓
1296 0A90 1005      JMP  INSR1A           NONE, USE POINTER.
1297 0A92 C801  MOV  UDCNT,@NUMB     FIND ADDR OF LINE AFTER WHICH
1298 0A94 013C ✓
1298 0A96 06A0      BL   @FLIN           INSERTS ARE TO BE MADE
1299 0A98 04E4 ✓
1299 0A9A C087  INSR1 MOV  TMLOC, LINAD
1300 0A9C C2A0  INSR1A MOV @HICOR, 10    IS THERE
1301 0A9E 0008 ✓
1301 0AA0 6284      S    LINPTR, 10      ROOM FOR
1302 0AA2 028A      CI   10, 80         INSERT?
1303 0AA4 0050
1303 0AA6 1B02      JH   $+6            YES
1304 0AA8 06A0      BL   @SQUEEZ        NO
1305 0AAA 0408 ✓
1305 0AAC C284      MOV  LINPTR, 10      CALC ADDR FOR
1306 0AAE 022A      AI   10, 6          INSERT
1307 0AB0 0006
1307 0AB2 C284      MOV  LINPTR, R10     CALC ADDR FOR
1308 0AB4 022A      AI   R10, 6         INSERT
1309 0AB6 0006
1309 0AB8 C30A      MOV  R10, R12        SAVE BUFFER STARTING ADDR
1310 0ABA 0420      BLWP @KEYIN         INPUT LINE
1311 0ABC 0FA4 ✓
1311 0ABE 06A0      MOV  @CR, *R10      END STRING WITH A CR
1312 0AC0 0185 ✓
1312 0AC2 830A      C    R10, R12        ANY INPUT?
1313 0AC4 131B      JEQ  EXTINS         NO
1314 0AC6 058A      INC  R10            SET ADDR
1315 0AC8 22A0      COC  @MARGLM, R10   FOR
1316 0ACA 012E ✓
1316 0ACC 1601      JNE  $+4            NEXT

```

1317	0ACE	058A	INC	R10	INSERT
1318	0AD0	C144	MOV	LINPTR, 5	
1319	0AD2	CD60	MOV	@INLNNO, *5+	SET LINE NUMBER
	0AD4	017A			
1320	0AD6	05A0	INC	@INLNNO	STEP INSERTED LINE COUNT
	0AD8	017A			
1321	0ADA	C302	MOV	LINAD, R12	
1322	0ADC	CD4C	MOV	R12, *5+	PUT ADDR OF PREV LINE IN HEAD
1323	0ADE	022C	AI	R12, 4	FIX ADDR OF WORD IN PREV LINE T
	0AE0	0004			
1324	0AE2	C25C	MOV	*R12, 9	FIND ADDR OF CELL IN NEXT
1325	0AE4	1302	JEQ	INSR6	LINE (IF IT EXISTS) TO
1326	0AE6	0509	INCT	9	BE UPDATED
1327	0AE8	0644	MOV	LINPTR, *9	UP-DATE NEXT-LINE TO POINT
1328	0AEA	050C	INSR6	MOV *R12, *R5	TO INSERTED LINE. MAKE INSERTE
1329	0AEC	0704	MOV	LINPTR, *R12	LINE POINT TO NEXT LINE. POINT
1330	0AEE	0208	MOV	CLINNO, CLINNO	PREV LINE TO INSERTED LINE
1331	0AF0	1602	JNE	INSR7	
1332	0AF2	0214	MOV	*LINPTR, CLINNO	IF POINTER ON DUMMY, MOVE
1333	0AF4	0190	MOV	*DUMNXT, CLLOC	TO TOP LINE IN BUFFER
1334			INSR7		
1335			*	PREPARE FOR NEXT INSERT	
1336	0AF6	C104	MOV	LINPTR, TMLOC	
1337	0AF8	C10A	MOV	R10, LINPTR	
1338	0AFA	10CF	JMP	INSR1	
1339	0AFC	C160	EXTINS	MOV @CHFLG, 5	
	0AFE	0116			
1340	0B00	1303	JEQ	*+0	
1341	0B02	C2E0	MOV	@SAVRTN, RTN	GET RETURN ADDR
	0B04	0154			
1342	0B06	045B	B	*RTN	GO BACK TO CHANGE
1343	0B08	0380	RTWP		

```
1346 * TITLE: KP
1347 * KEEP
1348 * REVISION:
1349 * ORIGINAL
1350 * COMPUTER: 990, ASSEMBLY
1351 * ABSTRACT: OUTPUTS SPECIFIED NUMBER OF SOURCE LINES
1352 * BEGINNING WITH TOP LINE
1353 * CALLING SEQUENCE:
1354 * BLWP @KP
1355 0B0A 0026' KP DATA ROUNTRG, $+2
0B0C 0B0E'
1356 0B0E 06A0 BL @SKANOP GET OPERAND
0B10 0500'
1357 0B12 1012 JMP DFLT DEFAULT KEEP COUNT
1358 0B14 C041 KP2 MOV UDCNT, UDCNT OPERAND ZERO?
1359 0B16 1602 JNE $+6 NO
1360 0B18 0460 B @OPNERR YES
0B1A 03E0'
1361 0B1C C1D0 MOV *DUMNXT, TMLOC
1362 0B1E 1602 JNE $+6
1363 0B20 0460 B @MEMPTY NOTHING TO SAVE
0B22 03EA'
1364 0B24 06A0 KEEP1 BL @SCROLL KEEP TOP LINE
0B26 0DCE'
1365 0B28 0601 DEC UDCNT FINISHED?
1366 0B2A 1309 JEQ MOEXT YES
1367 0B2C C1E0 MOV @NXTLAD, TMLOC GET POINTER TO
0B2E 014E'
1368 0B30 1602 JNE $+6 NEXT LINE
1369 0B32 0460 B @LSTLIN THERE IS NONE
0B34 03CC'
1370 0B36 10F6 JMP KEEP1 PROCESS NEXT LINE
1371 0B38 C060 DFLT MOV @LLIMIT, UDCNT
0B3A 0182'
1372 0B3C 10EB JMP KP2
1373 0B3E 06A0 MOEXT BL @SQUEEZ SQUEEZE UP BUFFER
0B40 0408'
1374 0B42 0380 RTWP
```

```

1377 * TITLE: LMTS
1378 * LIMITS COMMAND
1379 * REVISION:
1380 * ORIGINAL
1381 * COMPUTER: 990, ASSEMBLY
1382 * ABSTRACT: PRINTS THE FIRST AND LAST SOURCE LINES
1383 * IN BUFFER. PRINTS LINE NUMBER AND
1384 * ASSUMES FULL RIGHT HAND MARGINS
1385 * CALLING SEQUENCE:
1386 * BLWP @LMTS
1387 0B44 0026' LMTS DATA ROUTRG, $+2
      0B46 0B48'
1388 0B48 06A0 BL @SKANTM CHECK FOR CARRIAGE RETURN
      0B4A 051A'
1389 0B4C 0290 MOV *DUMNXT, 10 IS MEMORY EMPTY?
1390 0B4E 1602 JNE $+6 NO
1391 0B50 0460 B @MEMPTY YES
      0B52 03EA'
1392 0B54 0820 MOV @PLFLG, @SVPRM SAVE PLFLG AND
      0B56 011A'
      0B58 0160'
1393 0B5A 0820 MOV @PMRGIN, @SAVMRG PMRGIN LOCALLY,
      0B5C 0132'
      0B5E 0162'
1394 0B60 0820 MOV @MARGLM, @PLFLG THEN SET TEMPORARILLY
      0B62 012E'
      0B64 011A'
1395 0B66 0820 MOV @MARGLM+2, @PMRGIN
      0B68 0130'
      0B6A 0132'
1396 0B6C 06A0 BL @PRLIN PRINT TOP LINE
      0B6E 0068'
1397 0B70 06A0 BL @BOTTOM FIND ADDR OF BOTTOM LINE
      0B72 065A'
1398 0B74 0207 MOV IMLUC, 10
1399 0B76 06A0 BL @PRLIN PRINT IT
      0B78 0068'
1400 0B7A 0820 MOV @SVPRM, @PLFLG RESTORE PRINT
      0B7C 0160'
      0B7E 011A'
1401 0B80 0820 MOV @SAVMRG, @PMRGIN PARAMETERS
      0B82 0162'
      0B84 0132'
1402 0B86 0380 RTRN

```

```

1405 * TITLE: MV
1406 * MOVE COMMAND
1407 * REVISION:
1408 * ORIGINAL
1409 * COMPUTER: 990, ASSEMBLY
1410 * ABSTRACT: MOVES ONE OR MORE SOURCE LINES
1411 * TO ANOTHER LOCATION IN BUFFER
1412 * CALLING SEQUENCE:
1413 * BLWP @MV
1414 * STATISTICS: LINES ARE NOT PHYSICALLY MOVED, THE
1415 * APPROPRIATE POINTERS ARE CHANGED
1416 0B88 0026' MV DATA ROUTRG, $+2
      0B8A 0B8C'
1417 0B8C 06A0 BL @EXOP GET FIRST OPERAND
      0B8E 0030'
1418 0B90 0801 MOV UDCNT, @LML SAVE LINE-NO OF LAST MOVED LINE
      0B92 013A'
1419 0B94 0116 MOV CLLOC, TMLOC ASSUME NO SECOND OPERAND
1420 0B96 028C CI 12, >0D CARRIAGE RETURN?
      0B98 0000'
1421 0B9A 130E JEQ MV1 IF YES, USE POINTER
1422 0B9C 020C CI 12, >2C COMMA?
      0B9E 002C'
1423 0BA0 1649 JNE EREXMV NO ERROR
1424 0BA2 0807 MOV TMLOC, @MALIN
      0BA4 012A'
1425 0BA6 0801 MOV UDCNT, @MYRCNT SAVE UP-DOWN COUNT
      0BA8 0178'
1426 0BAH 06A0 BL @SKANOP GET SECOND OPERAND
      0BAC 0500'
1427 0BAE 1006 JMP MV1+4 NO NUMBER USE POINTER
1428 0BB0 0801 MOV UDCNT, @NUMB
      0BB2 013C'
1429 0BB4 06A0 BL @FLIN FIND ADDR OF LINE AFTER
      0BB6 04E4'
1430 0BB8 0807 MV1 MOV TMLOC, @MALIN WHICH LINES ARE TO BE MOVED
      0BBA 012A'
1431 0BBC 0143 MOV PLIMITS, 3 FIND ADDR OF LINE ABOVE STRNG TO B
1432 0BBE 0000 INCT 5 SAME AS LINE AFTER WHICH
1433 0BB8 0010 C *3, @MALIN STRING IS TO BE MOVED?
      0BB2 012A'
1434 0BB4 1330 JEQ EXTMV YES, NO MOVE REQ'D
1435 0BB6 0230 MOV *3, 10 NO
1436 0BB8 0160 MOV @MALIN, 5 FIND ADDR OF LINE THAT FOLLOWS
      0BBA 012A'
1437 0BCC 0220 HI 5, 4 STRING TO BE MOVED
      0BCE 0004'
1438 0BD0 0810 MOV *5, @LAMDV
      0BD2 012C'
1439 0BD4 0113 MOV PLIMITS, TMLOC FIND END OF STRING TO BE MOVED
1440 0BD6 0807 MV4 C TMLOC, @MALIN ERROR IF MOVING STRING AFTER
      0BD8 012A'
1441 0BD8 132C JEQ EREXMV ONE OF ITS MEMBERS
1442 0BDC 0160 MOV @ABRFLG, 5 ABSOLUTE OR

```

1443	0BE0	1609	JNE	MV5	RELATIVE?
1444	0BE2	8817	C	*TMLOC, @LML	END FOUND?
	0BE4	015A			
1445	0BE6	1309	JEQ	MV6	YES
1446	0BE8	0227	AI	TMLOC, 4	
	0BEH	0004			
1447	0BEC	C1D7	MOV	*TMLOC, TMLOC	
1448	0BEE	16F3	JNE	MV4	
1449	0BF0	0460	B	@NOTFND	
	0BF2	03F4			
1450	0BF4	0620	MV5	DEC	@MVRcnt
	0BF6	0178			
1451	0BF8	16F7	JNE	MV7	
1452	0BFA	C307	MV6	MOV	TMLOC, 12
1453	0BFC	0227	AI	TMLOC, 4	SAVE ADDR OF LINE AT END OF MVD ST GET ADDR OF LINE FOLLOWING
	0BFE	0004			
1454	0C00	C157	MOV	*TMLOC, 5	THE STRING TO BE MOVED
1455	0C02	1303	JEQ	MV6A	LINK THE LINES PRECEEDING AND FOLLOWING
1456	0C04	02C3	INCL	5	
1457	0C06	C54H	MOV	10, *5	THE MOVED
1458	0C08	0645	DECT	5	STRING
1459	0C0A	022A	MV6A	AI	10, 4
	0C0C	0004			
1460	0C0E	C685	MOV	5, *10	
1461	0C10	05C3	INCL	PLIMTS	LINK TOP OF MOVED STRING
1462	0C12	C1E0	MOV	@MALIN, TMLOC	TO THE LINE AFTER
	0C14	012A			
1463	0C16	C4C7	MOV	TMLOC, *PLIMTS	WHICH IT
1464	0C18	0227	AI	TMLOC, 4	WAS TO BE MOVED
	0C1A	0004			
1465	0C1C	0643	DECT	PLIMTS	BE MOVED
1466	0C1E	C5C3	MOV	PLIMTS, *TMLOC	
1467	0C20	C160	MOV	@LAMOV, 5	LINK BOTTOM OF THE MOVED STRING
	0C22	012C			
1468	0C24	1303	JEQ	MV6B	TO THE LINE FOLLOWING
1469	0C26	05C5	INCL	5	THE LINE AFTER
1470	0C28	C54C	MOV	12, *5	WHICH THE STRING
1471	0C2A	0645	DECT	5	WAS TO BE
1472	0C2C	022C	MV6B	AI	12, 4
	0C2E	0004			
1473	0C30	C705	MOV	5, *12	
1474	0C32	0380	EXTMV	RTWP	
1475	0C34	0460	EREXMV	B	@OPNERR
	0C36	03E0			ERROR EXIT

```

1478 * TITLE: PRNT
1479 * PRINT COMMAND
1480 * REVISION:
1481 * ORIGINAL
1482 * 03/15/76
1483 * MODIFIED TO RUN WITH PXRMT
1484 * COMPUTER: 990, ASSEMBLY
1485 * ABSTRACT: PRINTS LINES BETWEEN TWO LIMITS OR
1486 * RELATIVE TO THE CURRENT LINE POINTER
1487 * CALLING SEQUENCE:
1488 * BLWP @PRNT
1489 0C38 0026 PRNT DATA ROUTRG, #+2
      0C39 0030
1490 0C3C 06A0 BL @EXOP EXTRACT OPERANDS
      0C3E 0590
1491 0C40 C283 MOV PLIMITS, 10
1492 0C42 06A0 PRNT1 BL @PRLIN PRINT THE LINE
      0C44 0C68
1493 0C46 C820 MOV @ABRLFG, @ABRLFG ABSOLUTE OR RELATIVE?
      0C48 0110
      0C4A 0110
1494 0C4C 1308 JEQ PABS ABSOLUTE
1495 0C4E 0601 DEC UDCNT ALL LINES PRINTED?
1496 0C50 130A JEQ EXTPRT IF YES EXIT
1497 0C52 C2A0 PRNT2 MOV @SLINPT, 10 DOES NEXT LINE EXIST?
      0C54 0158
1498 0C56 C29A MOV *10, 10
1499 0C58 16F4 JNE PRNT1 IF YES PRINT ANOTHER ONE
1500 0C5A 0460 B @LSTLIN NO
      0C5C 0300
1501 0C5E C2H0 PABS MOV @LINADR, 10
      0C60 0146
1502 0C62 805H C *10, UDCNT ALL LINES PRINTED?
1503 0C64 16F6 JNE PRNT2 NO
1504 0C66 0380 EXTPRT RTWP
1505 * ROUTINE TO PRINT LINE
1506 * WORKSPACE ELEMENT 10 CONTAINS ADDR OF LINE TO BE PRINTED
1507 *
1508 * ROUTINE TO PRINT A LINE
1509 * IF TAB ENCOUNTERED, SPACES ARE PRINTED
1510 * TO NEXT TAB POSITION
1511 * LINE NUMBERS ARE PRINTED AS NEEDED
1512 * R10 CONTAINS ADDR OF LINE TO BE PRINTED
1513 *
1514 0C68 C80H PRLIN MOV R10, @LINADR SAVE LINE ADDR
      0C6A 0146
1515 0C6C 022H HI R10, 4 SAVE POINTER
      0C6E 0004
1516 0C70 C80H MOV R10, @SLINPT TO NEXT LINE
      0C72 0158
1517 0C74 050H INCT R10
1518 0C76 C80H MOV R10, @LINADR SAVE TEXT POINTER
      0C78 0144
1519 0C7A C146 MOV RTN, R5 SAVE RETURN

```

1520	0C7C	06H0		BL	@BDPRNT	PUT LINE NO
	0C7E	0C74				
1521	0C80	0260		MOV	@LINADD, R9	COPY ONE TEXT CHAR
	0C82	0144				
1522	0C84	0339	BLD1	MOVB	*R9+, R12	GET CHAR
1523	0C86	980C		CB	R12, @TAB	TAB?
	0C88	0198				
1524	0C8A	1320		JEQ	PRTB	YES
1525	0C8C	980C		CB	R12, @CR	CARRIAGE RETURN?
	0C8E	0185				
1526	0C90	1310		JEQ	BLD3	YES
1527	0C92	DA8C		MOVB	R12, @PBUF(R10)	MOVE ONE CHAR INTO BUFFER
	0C94	00C2				
1528	0C96	05A0		INC	@PRTPOS	INCR PRINT POSITION
	0C98	0188				
1529	0C9A	058A	BLD1A	INC	R10	INCR BUFF PTR
1530	0C9C	C2CA		MOV	R10, R11	SAVE CHAR COUNT
1531	0C9E	C820		MOV	@PLFLG, @PLFLG	LINE NOS BEING PRINTED?
	0CA0	011A				
	0CA2	011A				
1532	0CA4	1302		JEQ	BLD2	NO
1533	0CA6	022B		AI	R11, 6	ADJUST FOR THEM
	0CA8	0006				
1534	0CAA	880B	BLD2	C	R11, @PMARGIN	CHECK PRINT MARGIN
	0CAC	0132				
1535	0CAE	12EA		JLE	BLD1	TRANSFER ANOTHER CHAR
1536	0CB0	060A		DEC	R10	DECREMENT POINTER
1537	0CB2	DA80	BLD3	MOVB	@LF, @PBUF(R10)	TERM WITH LF/CR
	0CB4	0184				
	0CB6	00C2				
1538	0CB8	DA80		MOVB	@CR, @PBUF+1(R10)	
	0CBA	0185				
	0CBC	00C3				
1539	0CBE	C2A0		MOV	@PBUFA, R10	BUFFER ADDR FOR PRINT
	0CC0	00C0				
1540	0CC2	0420		BLWP	@PRINT	
	0CC4	0FEC				
1541	0CC6	04E0		CLR	@PRTPOS	REINIT PRINT POSITION
	0CC8	0188				
1542	0CCA	0455		B	*R5	RETURN
1543	0CCC	04CC	PRTB	CLR	R12	INSERT SPACES TO NEXT TAB
1544	0CCE	8B20	PRTB1	C	@PRTPOS, @TABS(R12)	FIND TAB GREATER THAN
	0CD0	0188				
	0CD2	018A				
1545	0CD4	1104		JLT	PRTSP	CURRENT PRINT POSITION
1546	0CD6	05CC		INCR	R12	INCR TAB BUFF INDEX
1547	0CD8	028C		CI	R12, 8	UP TO 4 TABS ALLOWED
	0CDA	0008				
1548	0CDC	16F8		JNE	PRTB1	
1549	0CDE	DA80	PRTSP	MOVB	@SP, @PBUF(R10)	FOUND, INSERT SPACES
	0CE0	0186				
	0CE2	00C2				
1550	0CE4	05A0		INC	@PRTPOS	INCR PRINT POSITION
	0CE6	0188				
1551	0CE8	8B20		C	@PRTPOS, @TABS(R12)	UNTIL TAB STOP REACHED


```

00CEA 0188/
00CEC 018A/
1552 00CEE 14D5      JHE  BLD1A      GO CHECK PRINT LIMITS
1553 00CF0 058A      INC  R10        UP TO NEXT TAB STOP
1554 00CF2 10F5      JMP  PRTSP
1555      * CONVERT BINARY LINE NUMBER TO DECIMAL ASCII
1556 00CF4 C2A0      BDPRNT MOV @PLFLG,10      PRINT LINE NUMBERS?
00CF6 011A/
1557 00CF8 04CA      CLR  10
1558 00CFA 1327      JEQ  EXTBP1     EXIT IF NO NUMBERS TO PRINTED
1559 00CFC C2A0      MOV  @LINADR,10
00CFE 0146/
1560 0D00 C25A      MOV  *10,9      WAS LINE INSERTED?
1561 0D02 1112      JLT  PUTSPS     IF YES FILL BUFFER WITH SPACES
1562      *
1563      * CONVERT BINARY NUMBER IN R9 TO DECIMAL ASCII
1564      *
1565      CONVRT
1566 0D04 020C      LI   R12,4
0D06 0004
1567      CNVRT
1568 0D08 C289      MOV  R9,R10     BINARY NUMBER RIGHT
1569 0D0A 04C9      CLR  R9         JUSTIFIED IN R9 & R10
1570 0D0C 3E60      DIV  @TEN,R9
0D0E 017C/
1571 0D10 0B8A      SRC  R10,8      LEFT JUSTIFY CHAR
1572 0D12 DB0A      MOVB R10,@PBUF(12) STORE CHAR
0D14 00C2/
1573 0D16 BB20      AB   @BINASC,@PBUF(R12)
0D18 0194/
0D1A 00C2/
1574 0D1C 060C      DEC  R12
1575 0D1E 15F4      JGT  CNVRT
1576 0D20 D820      MOVB @SP,@PBUF  FIRST CHAR IS SPACE
0D22 0186/
0D24 00C2/
1577 0D26 1007      JMP  EXTBDP
1578      PUTSPS
1579 0D28 020A      LI   R10,4      PUT SPACES FOR LINE NUMBER
0D2A 0004
1580      PTSP1
1581 0D2C DAA0      MOVB @SP,@PBUF(R10)
0D2E 0186/
0D30 00C2/
1582 0D32 060A      DEC  R10
1583 0D34 18FB      JOC  PTSP1
1584      EXTBDP
1585 0D36 D820      MOVB @SP,@PBUF+5
0D38 0186/
0D3A 00C7/
1586 0D3C D820      MOVB @CR,@PBUF+6
0D3E 0185/
0D40 00C8/
1587 0D42 C2A0      MOV  @PBUFA,R10 SET UP BUFFER ADDR
0D44 00C0/

```

PXREDT
PRINT COMMAND

MIRA990 V2L1 12:13:11
948927-9901**

258/76

PAGE 0048

1588	0D46	0420	BLWP	@PRINTN
	0D48	0FF0		
1589			EXTBP1	
1590	0D4A	04CA	CLR	R10
1591	0D4C	045B	B	*RTN

PRINT LINE

```
1594 * TITLE: QT
1595 * QUIT COMMAND
1596 * REVISION:
1597 * ORIGINAL
1598 * COMPUTER: 990, 990 ASSEMBLY
1599 * ABSTRACT: TERMINATES THE EDIT PROCESS AFTER COPYING
1600 * THE REMAINDER OF THE SPECIFIED DATA ONTO THE
1601 * OUTPUT FILE.
1602 *
1603 * @ COPIES REMAINDER OF BUFFER AND INPUT
1604 * SOURCE FILE TO OUTPUT FILE
1605 * @<S> COPIES REMAINDER OF BUFFER AND INPUT
1606 * SOURCE FILE TO LINE NUMBER <S> TO
1607 * OUTPUT FILE.
1608 * @ @ COPIES REMAINDER OF BUFFER TO OUTPUT FIL
1609 *
1610 * THE FLAG EXTFLG IS SET TO INDICATE TO THE MAIN
1611 * ROUTINE TO WRITE AN END OF FILE RECORD AND
1612 * RESTART THE EDITOR.
1613 * CALLING SEQUENCE:
1614 * BLWP @QT
1615 *
1616 0D4E 0026' QT DATA ROUNTRG, #+2
1617 0D50 0D52' CLR @FINLAD ASSUME NO OPERAND
1618 0D54 014C'
1619 0D56 06A0 BL @SKANOP GET OPERAND IF ANY
1620 0D58 0500'
1621 0D5A 100F JMP QT2 NONE, SAVE EVERYTHING.
1622 0D5C C241 MOV UDCNT, 9 IS IT ZERO
1623 0D5E 1603 JNE QT0 NO
1624 0D60 05A0 INC @EOFLG YES, SET UP FOR LAST LINE
1625 0D62 0114'
1626 0D64 100A JMP QT2
1627 0D66 0809 MOV 9, @NUMB
1628 0D68 013C'
1629 0D6A 0707 SETO TMLOC ASSUME LINE NOT IN BUFFER
1630 0D6C 8809 C 9, @MAXLIN HAS THE LINE BEEN IN CORE?
1631 0D6E 015C'
1632 0D70 1502 JGT QT1 NO
1633 0D72 06A0 BL @FLIN YES, SEE IF IT IS STILL THERE.
1634 0D74 04E4'
1635 0D76 C807 QT1 MOV TMLOC, @FINLAD
1636 0D78 014C'
1637 0D7A C1D0 QT2 MOV *DUMNXT, TMLOC IS MEMORY EMPTY?
1638 0D7C 160B JNE QT3 NO
1639 0D7E C320 MOV @EOFLG, 12 YES, END OF FILE SET.
1640 0D80 0114'
1641 0D82 1611 JNE QT5 IF YES, LINE NOT FOUND
1642 0D84 C060 MOV @MAXLIN, UDCNT
1643 0D86 015C'
1644 0D88 0221 AI UDCNT, 10
1645 0D8A 000A
```

```
1637 0D8C 06A0          BL  @INSOU          BRING IN TEN LINES
      0D8E 0708
1638 0D90 1000          JMP  #+2           EOF EXIT
1639 0D92 10F3          JMP  QT2
1640 0D94 06A0  QT3      BL  @SCROLL        SCROLL TOP LINE
      0D96 0DCE
1641 0D98 C320          MOV  @FINLAD, 12   CHECK FLAG
      0D9A 014C
1642 0D9C 130B          JEQ  SAVALL        SCROLL EVERYTHING
1643 0D9E 1110          JLT  QT4           LOOK AT CURNO
1644 0DA0 808C          C    12, LINAD     LAST LINE GONE?
1645 0DA2 1312          JEQ  EXTQT         YES
1646 0DA4 10EA          JMP  QT2
1647 0DA6 C320  QT5      MOV  @FINLAD, 12   SAVE ALL?
      0DA8 014C
1648 0DAA 130E          JEQ  EXTQT         YES
1649 0DAC 06A0          BL  @MSGOUT        NO
      0DAE 02CE
1650 0DB0 0010          DATA 16          LAST LINE
1651 0DB2 100A          JMP  EXTQT
1652 0DB4 C1D0  SAVALL    MOV  *DUMNXT, TMLDC IS MEMORY EMPTY?
1653 0DB6 16EE          JNE  QT3           NO, SCROLL OUT MORE.
1654 0DB8 C320          MOV  @EOFLG, 12   END OF FILE FOUND
      0DBA 0114
1655 0DBC 13DE          JEQ  QT2           NO, SCROLL OUT MORE.
1656 0DBE 1004          JMP  EXTQT         YES
1657 0DC0 8820  QT4      C    @CURNO, @NUMB LAST LINE OUT?
      0DC2 0150
      0DC4 013C
1658 0DC6 16D9          JNE  QT2           IF NO, SCROLL OUT MORE
1659 0DC8 0720  EXTQT    SETO @EXTFLG     YES, SET QUIT FLAG & EXIT
      0DCA 0118
1660 0DCC 0380          RTWP
```

```

1663 * TITLE: SCROLL/KLRLIN
1664 * SCROLL LINE FROM MEMORY/CLEAR LINE HEADER
1665 * REVISION:
1666 * ORIGINAL
1667 * 03/15/76
1668 * MODIFIED TO RUN WITH PXRMTN
1669 * COMPUTER: 990, ASSEMBLY
1670 * ABSTRACT: SCROLL WILL SCROLL A LINE OUT OF MEMORY,
1671 * TMLOC CONTAINS ADDR OF LINE TO BE SCROLLED
1672 * SCROLL FALLS THROUGH TO KLRLIN.
1673 * KLRLIN CLEARS THE HEADER OF THE LINE WHOSE
1674 * ADDRESS IS IN REGISTER 5.
1675 * CALLING SEQUENCE:
1676 * BL @SCROLL
1677 * BL @KLRLIN
1678 SCROLL
1679 0DCE C287 MOV TMLOC,R10 SET BUFFER ADDR FOR OUTPUT
1680 0DD0 022A AI R10,6 INCR PAST HEADER INFO
1681 0DD2 0006
1681 0DD4 0420 BLWP @WRITE WRITE IT OUT
1681 0DD6 0FE0'
1682 0DD8 1309 JEQ SCR0 IF NO ERROR
1683 0DDA 020A LI R10,MSG14 PRINT RDY OBJ-TYPE CR
1683 0DDC 03BA'
1684 0DDE 0420 BLWP @PRINTN
1684 0DE0 0FF0'
1685 0DE2 C2A0 MOV @PBUFA,R10 INPUT RESPONSE
1685 0DE4 00C0'
1686 0DE6 0420 BLWP @KEYIN
1686 0DE8 0FA4'
1687 0DEA 10F1 JMP SCROLL CONTINUE
1688
1688 0DEC C087 SCR0 MOV TMLOC,LINAD ADDR OF LINE TO BE MOVED OUT
1689 0DEE C837 MOV *TMLOC+,@CURNO CURRENT LINE NUMBER
1689 0DF0 0150'
1691 0DF2 C807 MOV TMLOC,@PRLAD PREVIOUS LINE POINTER
1691 0DF4 0152'
1692 0DF6 05C7 INCT TMLOC
1693 0DF8 C817 MOV *TMLOC,@NXTLAD ADDR OF NEXT LINE
1693 0DFA 014E'
1694 0DFC C417 MOV *TMLOC,*DUMNXT IS THERE A NEXT LINE?
1695 0DFE 1305 JEQ SCR1 YES
1696 0E00 C257 MOV *TMLOC,9
1697 0E02 05C9 INCT 9
1698 0E04 C650 MOV @ANCHOR,*9
1698 0E06 017E'
1699 0E08 1004 JMP SCR2
1700 0E0A C120 SCR1 MOV @INTLPT,LINPTR RESET LINE POINTER TO TOP
1700 0E0C 0180'
1701 0E0E 0224 AI LINPTR,4
1701 0E10 0004
1702 0E12 05C7 SCR2 INCT TMLOC
1703 0E14 C142 MOV LINAD,5 SAVE ADDR OF LINE
1704 *

```

```

1705 * THIS ROUTINE CLEARS THE HEADER OF THE LINE WHOSE
1706 * ADDR IS IN ELEMENT 5
1707 *
1708 0E16 0735 KLRLIN SETO *5+ REPLACE LINE NUMBER WITH -1
1709 0E18 04F5 CLR *5+ REPLACE POINTERS
1710 0E1A 04D5 CLR *5 WITH ZERO
1711 0E1C 8220 C @CURNO,CLINNO HAS LINE AT POINTER BEEN REMOVE
1712 0E1E 0150
1713 0E20 1604 JNE EXTSCR NO
1714 0E22 C1D0 MOV *DUMNXT, TMLOC YES, SET POINTERS.
1715 0E24 1303 JEQ SCR3 MEMORY EMPTY?
1716 0E26 C187 MOV TMLOC, CLLOC NO
1717 0E28 C217 SCR4 MOV *TMLOC, CLINNO
1718 0E2A 045B EXTSCR B *RTN EXIT SCROLL
1719 0E2C C1A0 SCR3 MOV @ANCHOR, CLLOC YES
1720 0E2E 017E
1721 0E30 C207 MOV TMLOC, CLINNO
1722 0E32 10FB JMP EXTSCR

```

```

1723 * TITLE: RMV
1724 * REMOVE COMMAND
1725 * REVISION:
1726 * ORIGINAL
1727 * COMPUTER: 990. ASSEMBLY
1728 *
1729 * ABSTRACT: INACTIVATES SOURCE BY SETTING LINE
1730 * NUMBER TO -1 AND ZEROING REST OF HEADER
1731 * CALLING SEQUENCE:
1732 * BLWP @RMV
1733 * OR
1734 * BL @RMVCHR
1735 0E34 0026' RMV DATA ROUTRG, RMV6
      0E36 0E3C'
1736 0E38 C80B RMVCH MOV RTN, @SAVRTN ENTRY POINT WHEN CALLED BY C
      0E3A 0154'
1737 0E3C 06A0 RMV6 BL @EXOP GET OPERANDS
      0E3E 0590'
1738 0E40 C160 MOV @CHFLG, 5 CALLED BY CHANGE?
      0E42 0116'
1739 0E44 1304 JEQ RMV1 NO
1740 0E46 C1C3 MOV PLIMTS, TMLOC YES, CALC ADDR FOR INSRT.
1741 0E48 05C7 INCT TMLOC AND PUT IT BACK
1742 0E4A C817 MOV *TMLOC, @CHFLG IN CHANGE FLAG
      0E4C 0116'
1743 0E4E C083 RMV1 MOV PLIMTS, LINAD
1744 0E50 C833 MOV *PLIMTS+, @CURNO SAVE LINE NUMBER
      0E52 0150'
1745 0E54 C803 MOV PLIMTS, @PRLAD
      0E56 0152'
1746 0E58 C833 MOV *PLIMTS+, @PRELAD SAVE ADDR OF PREV LINE
      0E5A 0148'
1747 0E5C C803 MOV PLIMTS, @NXLAD
      0E5E 014A'
1748 0E60 C813 MOV *PLIMTS, @NXTLAD SAVE ADDR OF NEXT LINE
      0E62 014E'
1749 0E64 C1E0 MOV @PRELAD, TMLOC
      0E66 0148'
1750 0E68 8807 C TMLOC, @ANCHOR IS THERE A PREVIOUS LINE?
      0E6A 017E'
1751 0E6C 130C JEQ RMV2 NO
1752 0E6E 0227 AI TMLOC, 4 YES, LINK WITH IT.
      0E70 0004
1753 0E72 C160 MOV @NXTLAD, 5 NEXT LINE
      0E74 014E'
1754 0E76 C5C5 MOV 5, *TMLOC
1755 0E78 130A JEQ CLRLIN
1756 0E7A 05C5 RMV4 INCT 5 LINK NEXT
1757 0E7C C1C5 MOV 5, TMLOC LINE TO
1758 0E7E C160 MOV @PRELAD, 5 PREVIOUS LINE
      0E80 0148'
1759 0E82 C5C5 MOV 5, *TMLOC
1760 0E84 1004 JMP CLRLIN GO CLEAR THIS LINE
1761 * THE LINE BEING REMOVED IS THE TOP ONE

```

1762	0E86	C160	RMV2	MOV	@NXTLAD, 5	
	0E88	014E				
1763	0E8A	C405		MOV	5, *DUMNXT	
1764	0E8C	16F6		JNE	RMV4	
1765	0E8E	C142	CLRLIN	MOV	LINAD, 5	
1766	0E90	06A0		BL	@KLRLIN	CLEAR THE LINE
	0E92	0E16				
1767	0E94	C160		MOV	@ABRLFG, 5	RELATIVE OR ABSOLUTE?
	0E96	011C				
1768	0E98	1604		JNE	RELRMV	RELATIVE
1769	0E9A	8060		C	@CURNO, UDCNT	LAST LINE REMOVED?
	0E9C	0150				
1770	0E9E	1603		JNE	REXB	NO
1771	0EA0	1007		JMP	EXTRMV	YES
1772	0EA2	0601	RELRMV	DEC	UDCNT	
1773	0EA4	1305		JEQ	EXTRMV	
1774	0EA6	C0E0	REXB	MOV	@NXTLAD, PLIMTS	GET POINTER
	0EA8	014E				
1775	0EAA	1601		JNE	RMV1	TO NEXT LINE
1776	0EAC	0460		B	@LSTLIN	THERE IS NONE
	0EAE	03CC				
1777	0EB0	C160	EXTRMV	MOV	@CHFLG, 5	CALLED BY CHANGE?
	0EB2	0116				
1778	0EB4	1601		JNE	CHRET	
1779	0EB6	0380		RTWP		
1780	0EB8	C2E0	CHRET	MOV	@SAVRTN, RTN	RETURN TO
	0EBA	0154				
1781	0EBC	045B		B	*RTN	CHANGE


```

1784 * TITLE: STP
1785 * SETUP COMMANDS
1786 * REVISION:
1787 * ORIGINAL
1788 * COMPUTER: 990, ASSEMBLY
1789 * ABSTRACT: SETS UP PRINT LIMITS, FIND MARGIN,
1790 * PRINT LINE NUMBERS, AND PRINT NO
1791 * LINE NUMBERS.
1792 * CALLING SEQUENCE:
1793 * BLWP @STP
1794 0EBE 0026' STP DATA ROUTRG, $+2
      0EC0 0EC2'
1795 0EC2 06A0 BL @NUMEXT
      0EC4 0526'
1796 0EC6 028C CI 12, 'L'
      0EC8 004C
1797 0ECA 1310 JEQ SLIN SET PRINT LINE NUMBER FLAG
1798 0ECC 028C CI 12, 'P'
      0ECE 0050
1799 0ED0 1312 JEQ SPLIM SET UP PRINT MARGINS
1800 0ED2 028C CI 12, 'N'
      0ED4 004E
1801 0ED6 1305 JEQ SNLIN SETUP TO NOT PRINT LINE NUMBER
1802 0ED8 028C CI 12, 'M'
      0EDA 004D
1803 0EDC 131A JEQ SMARG SET FIND MARGIN
1804 0EDE 0460 B @OPNERR ANYTHING ELSE ILLEGAL
      0EE0 03E0'
1805 0EE2 06A0 SNLIN BL @SKANTM CLEAR PRINT-
      0EE4 051A'
1806 0EE6 04E0 CLR @PLFLG LINE-NUMBER
      0EE8 011A'
1807 0EEA 1026 JMP EXTSTP FLAG
1808 0EEC 06A0 SLIN BL @SKANTM SET PRINT-
      0EEE 051A'
1809 0EF0 0720 SETD @PLFLG LINE-NUMBER
      0EF2 011A'
1810 0EF4 1021 JMP EXTSTP FLAG
1811 0EF6 C1E0 SPLIM MOV @MARGLM+2, TMLOC
      0EF8 0130'
1812 0EFA 06A0 BL @SKANOP GET NO OF CHARS TO PRINT
      0EFC 0500'
1813 0EFE C047 MOV TMLOC, UDCNT
1814 0F00 0281 CI UDCNT, 10
      0F02 000A
1815 0F04 111A JLT EREXST TOO SMALL
1816 0F06 0281 CI UDCNT, 80
      0F08 0050
1817 0F0A 1517 JGT EREXST TOO LARGE
1818 0F0C C801 MOV UDCNT, @PMARGIN SAVE CHAR COUNT
      0F0E 0132'
1819 0F10 1013 JMP EXTSTP
1820 0F12 06A0 SMARG BL @SKANOP GET LOW MARGIN
      0F14 0500'
  
```

1821	0F16	1013	JMP	MDFLT	
1822	0F18	C801	MOV	UDCNT, @LOMRG	ZERO NOT
	0F1A	0134'			
1823	0F1C	130E	JEQ	EREXST	ALLOWED
1824	0F1E	028C	CI	12, ', '	COMMA IS THE ONLY
	0F20	002C			
1825	0F22	160B	JNE	EREXST	VALID DELIMETER
1826	0F24	06A0	BL	@NUMEXT	GET HIGH MARGIN
	0F26	0526'			
1827	0F28	C160	MOV	@NUMBF, 5	
	0F2A	013A'			
1828	0F2C	1306	JEQ	EREXST	NO NUMBER
1829	0F2E	C801	MOV	UDCNT, @HIMRG	SAVE HIMRG
	0F30	0136'			
1830	0F32	8060	C	@LOMRG, UDCNT	
	0F34	0134'			
1831	0F36	1401	JHE	EREXST	LOMRG >= HIMRG.
1832	0F38	0380	EXTSTP	RTWP	
1833	0F3A	0460	EREXST	B @OPNERR	
	0F3C	03E0'			
1834	0F3E	C820	MDFLT	MOV @MARGLM, @LOMRG	DEFAULT - LEFT MARGIN = 1
	0F40	012E'			
	0F42	0134'			
1835	0F44	C820	MOV	@MARGLM+2, @HIMRG	RIGHT MARGIN = PXR
	0F46	0130'			
	0F48	0136'			
1836	0F4A	0380	RTWP		

```
1839 * TITLE: TOP
1840 * TOP COMMAND
1841 * REVISION:
1842 * ORIGINAL
1843 * COMPUTER: 990, ASSEMBLY
1844 * ABSTRACT: MOVE POINTER TO FIRST ACTIVE LINE
1845 * IN BUFFER
1846 * CALLING SEQUENCE:
1847 * BLWP @TOP
1848 0F4C 0026' TOP DATA ROUTRG, $+2
      0F4E 0F50'
1849 0F50 06A0 BL @SKANTM CHECK TERM CHAR
      0F52 051A'
1850 0F54 C1D0 MOV *DUMNXT, TMLOC IS MEMORY EMPTY?
1851 0F56 1303 JEQ TP1 YES
1852 0F58 C187 MOV TMLOC, CLLOC NO, SET POINTERS.
1853 0F5A C217 MOV *TMLOC, CLINNO
1854 0F5C 0380 RTWP
1855 0F5E C1A0 TP1 MOV @ANCHOR, CLLOC
      0F60 017E'
1856 0F62 C217 MOV *TMLOC, CLINNO
1857 0F64 0460 B @MEMPTY
      0F66 03EA'
```

```
1860 *
1861 * TITLE: OPEN
1862 * ASSIGN A DEVICE TO A TASK
1863 * REVISION: 03/15/76
1864 * ORIGINAL
1865 * COMPUTER: 990, ASSEMBLY
1866 * ABSTRACT: SETS UP PRB AND MAKES SUPERVISOR
1867 * CALL TO OPEN A DEVICE
1868 * CALLING SEQUENCE:
1869 * R10 = LUN0
1870 * BLWP @OPEN
1871 OPEN
1872 0F68 004C/ DATA IOWKS, #+2 TRANSFER VECTOR
    0F6A 0F6C/
1873 0F6C 04C3 CLR IOPLUN
1874 0F6E D0E0 MOVB @OPNCOD, IOPLUN STORE I/O OP
    0F70 01A4/
1875 0F72 04C4 CLR FLGS INIT FLAGS
1876 0F74 04C2 CLR IOC INIT CODE FOR I/O SUPV CALL
1877 0F76 C30D MOV R13, R12 GET CALLER'S WORKSPACE PTR
1878 0F78 022C AI R12, R10+R10 INDEX TO CALL PARAMETER
    0F7A 0014
1879 0F7C C31C MOV *R12, R12 GET LUN0
1880 0F7E E0CC SOC R12, IOPLUN OR LUN0 INTO IOPLUN
1881 0F80 2FC2 SVC IOC
1882 0F82 0380 RTWP RETURN
```

* TITLE: ENDFIL
* WRITE END OF FILE
* REVISION: 03/15/76
* ORIGINAL
* COMPUTER: 990, ASSEMBLY
* ABSTRACT: SETS UP PRB AND MAKES SUPV CALL
* TO WRITE END OF FILE
* CALLING SEQUENCE:
* BLWP @ENDFIL

ENDFIL

1884	0F84	004C	DATA	IOWKS, #+2	TRANSFER VECTOR
1885	0F86	0F88			
1886	0F88	04C2	CLR	IOC	INIT CODE FOR I/O SUPV CALL
1887	0F8A	00E0	MOV	@C2LUN0, IOPLUN	STORE LUN0
1888	0F8C	01A0			
1889	0F8E	D0E0	MOVB	@E0FC0D, IOPLUN	STORE I/O OP
1890	0F90	01A5			
1891	0F92	04C4	CLR	FLGS	INIT FLAGS
1892	0F94	04C7	CLR	CHRCNT	INIT CHAR CNT TO ZERO
1893	0F96	2FC2	SVC	IOC	MAKE SUPV CALL
1894	0F98	0380	RTWP		RETURN

```
1903 * TITLE: READ
1904 * READ A RECORD
1905 * REVISION: 03/15/76
1906 * ORIGINAL
1907 * COMPUTER: 990, ASSEMBLY
1908 * ABSTRACT: SETS UP PRB AND MAKES SUPV CALL
1909 * TO READ A RECORD
1910 * CALLING SEQUENCE:
1911 * R10 = BUFFER ADDRESS
1912 * BLWP @READ
1913 * STATISTICS:
1914 * RETURN PARAMETERS:
1915 * R10 = BUFFER ADDR+CHAR COUNT
1916 * R9 = FLAGS
1917 READ
1918 0F9A 004C' DATA IOWKS, $+2 TRANSFER VECTOR
1919 0F9C 0F9E'
1919 0F9E C0E0 MOV @C1LUN0, IOPLUN STORE LUN0
1919 0FA0 019E'
1920 0FA2 1004 JMP KEYRD
```

```

1922 * TITLE: KEYIN
1923 * KEYBOARD INPUT ROUTINE
1924 * REVISION: 03/15/76
1925 * ORIGINAL
1926 * COMPUTER: 990, ASSEMBLY
1927 * ABSTRACT: SETS UP PRB AND MAKES SUPV CALL
1928 * TO INPUT A RECORD FROM KEYBOARD
1929 * CALLING SEQUENCE:
1930 * R10 = BUFFER ADDRESS
1931 * BLWP @KEYIN
1932 * STATISTICS:
1933 * RETURN PARAMETERS:
1934 * R10 = BUFFER ADDRESS+CHAR COUNT
1935 * R9 = FLAGS
1936 KEYIN
1937 0FA4 004C' DATA IOWKS, $+2 TRANSFER VECTOR
1938 0FA6 0FAB'
1939 0FA8 C0E0 MOV @KBLUNO, IOPLUN STORE LUNO
1940 0FAA 019A'
1941 KEYRD
1942 0FAC D0C0 MOV B RDCODP, IOPLUN STORE I/O OP
1943 0FAE C30D MOV R13, R12 GET CALLER'S WORKSPACE PTR
1944 0FB0 022C AI R12, R10+R10 INDEX TO CALLING PARAMETERS
1945 0FB2 0014
1946 0FB4 C15C MOV *R12, BUFADR SET UP BUFFER ADDRESS
1947 0FB6 C18A MOV LEN, BUFLN SET UP BUFFER LENGTH
1948 0FB8 04C4 CLR FLGS INIT FLAGS
1949 0FBA 04C7 CLR CHRCNT INIT CHAR CNT TO ZERO
1950 0FBC 04C2 CLR IOC INIT CODE FOR I/O SUPV CODE
1951 0FBE 2FC2 SVC IOC MAKE SUPV CALL
1952 0FC0 A707 A CHRCNT, *R12 INCR BUFF PTR TO END OF DATA
1953 0FC2 0243 ANDI IOPLUN, >00FF IS THIS KEYBOARD READ?
1954 0FC4 00FF
1955 0FC6 8803 C IOPLUN, @KBLUNO
1956 0FC8 019A'
1957 0FCA 1303 JEQ KEYPLF IF YES, GO PRINT LF
1958 0FCC 064C DECT R12
1959 0FCE C704 MOV FLGS, *R12 MOVE FLAGS TO CALLING WS
1960 0FD0 0380 RTWP ELSE, RETURN
1961 KEYPLF
1962 0FD2 C0E0 MOV @PRLUNO, IOPLUN SET UP PRB TO PRINT LF
1963 0FD4 019C'
1964 0FD6 D0C1 MOV B WRCODP, IOPLUN
1965 0FD8 0205 LI BUFADR, LFCR SET UP BUFFER ADDR
1966 0FDA 0184'
1967 0FDC 0704 SETO FLGS SET FLAG TO PRINT WITHOUT CR
1968 0FDE 1014 JMP PRTKEY JUMP TO ENTRY POINT IN PRINT

```

```
1963 * TITLE: WRITE
1964 * WRITE A RECORD
1965 * REVISION: 03/15/76
1966 * ORIGINAL
1967 * COMPUTER: 990, ASSEMBLY
1968 * ABSTRACT: SETS UP PRB AND MAKES SUPV CALL
1969 * TO WRITE A RECORD
1970 * CALLING SEQUENCE:
1971 * R10 = BUFFER ADDRESS
1972 * BLWP @WRITE
1973 WRITE
1974 0FE0 004C' DATA IOWKS, $+2 TRANSFER VECTOR
      0FE2 0FE4'
1975 0FE4 C0E0 MOV @C2LUNO, IOPLUN STORE LUNO
      0FE6 01A0'
1976 0FE8 0704 SETO FLGS SET FLAG FOR WRITE WITHOUT CR
1977 0FEA 1009 JMP PRWRT
```



```

079      * TITLE:   PRINT/PRINTN
080      *          PRINT A RECORD
1981     * REVISION: 03/15/76
1982     *          ORIGINAL
1983     * COMPUTER: 990, ASSEMBLY
1984     * ABSTRACT: SETS UP PRB AND MAKES SUPV CALL
1985     *          TO PRINT A RECORD
1986     *          PRINT - PRINTS RECORD WITH CR
1987     *          PRINTN- PRINTS RECORD WITHOUT CR
1988     * CALLING SEQUENCE:
1989     *          R10 = BUFFER ADDRESS
1990     *          BLWP @PRINT
1991     *          OR
1992     *          BLWP @PRINTN
1993     PRINT
1994     0FEC  004C'  DATA IOWKS, PRTCR      TRANSFER VECTOR
           0FEE  0FF8'
1995     PRINTN
1996     0FF0  004C'  DATA IOWKS, PRTNCR     TRANSFER VECTOR
           0FF2  0FF4'
1997     PRTNCR
1998     0FF4  0704   SETO FLGS          SET FLAG FOR PRINT WITHOUT CR
1999     0FF6  1001   JMP PRENT
2000     PRTCR
2001     0FF8  04C4   CLR  FLGS          SET FLAG FOR PRINT WITH CR
2002     PRENT
2003     0FFA  C0E0   MOV  @PRLUN0, IOPLUN     STORE LUN0
           0FFC  019C'
2004     PRWRT
2005     0FFE  D0C1   MOVB WRCODP, IOPLUN     STORE I/O OP
2006     1000  C30D   MOV  R13, R12          GET CALLER'S WP
2007     1002  022C   AI   R12, R10+R10     INDEX TO CALLING PARAMETERS
           1004  0014
2008     1006  C15C   MOV  *R12, BUFADR      SET UP BUFFER ADDR
2009     PRTKEY
2010     1008  06A0   BL   @GETCNT          GET CHAR COUNT
           100A  102C'
2011     100C  A1C4   A    FLGS, CHRCNT     ADJUST CHRCNT FOR CR
2012     100E  04C4   CLR  FLGS          INIT FLAGS
2013     1010  04C2   CLR  IOC          INIT CODE FOR I/O SUPV CALL
2014     1012  2FC2   SVC  IOC          MAKE SUPV CALL
2015     1014  0244   ANDI FLGS, IOERR     CHECK RETURN STATUS
           1016  4000
2016     1018  02CF   STST R15          STORE FLAGS IN USER STATUS
2017     101A  1307   JEQ  PREXT          EXIT IF NO ERROR
2018     101C  0243   ANDI IOPLUN, >00FF
           101E  00FF
2019     1020  8803   C    IOPLUN, @PRLUN0 PRINT ERR (ESC CNTL RTN)
           1022  019C'
2020     1024  1602   JNE  PREXT          NO, WRITE ERR (TAPE NR)
2021     1026  0460   B    @ESCRTN       RETURN TO MAIN LOOP
           1028  021C'
2022     PREXT
2023     102A  0380   RTWP          RETURN

```

```

2024 * TITLE: GETCNT
2025 * GET CHARACTER COUNT
2026 * REVISION: 03/15/76
2027 * ORIGINAL
2028 * COMPUTER: 990, ASSEMBLY
2029 * ABSTRACT: SCANS LINE COUNTING CHARACTERS UP TO
2030 * AND INCLUDING A CARRIAGE RETURN
2031 * CALLING SEQUENCE:
2032 * R10 = BUFFER ADDRESS
2033 * BL @GETCNT
2034 * STATISTICS:
2035 * RETURN PARAMETERS:
2036 * CHRCNT(R7) = CHARACTER COUNT
2037 * CALLING SEQUENCE:
2038 * BL @GETCNT
2039 GETCNT
2040 102C 0407 CLR CHRCNT INIT CHAR COUNT
2041 GCLOOP
2042 102E 0587 INC CHRCNT INCR CHAR COUNT
2043 1030 9835 CB *BUFADR+, @CR COMPARE CHAR TO CR
2044 1032 0185 JEQ GCEXT IF EQUAL, EXIT
2045 1036 0287 CI CHRCNT, 82 MORE THAN 80 CHARS + LF/CR?
2046 1038 0052
2047 103A 11F9 JLT GCLOOP NO, CONTINUE
2048 103C 6147 GCEXT S CHRCNT, BUFADR RESET BUFFER ADDR TO START
2049 103E 045B B *RTN RETURN

```

```

0052 * TITLE: PGEND/INIT
0053 * BUFFER AREA/HEADING & MEMSIZ INIT
2054 * REVISION:
2055 * ORIGINAL
2056 * 03/15/76
2057 * HEADING AND MEM SIZE INIT MOVED &
2058 * MODIFIED TO RUN WITH PXRMTN
2059 * 12/07/76
2060 * MODIFIED TO RUN WHEN BOOTED WITH PXR90 MONITOR
2061 * COMPUTER: 990, ASSEMBLY
2062 * ABSTRACT: AREA FROM END OF EDITOR TO 4K BOUNDARY
2063 * RESERVED FOR EDIT BUFFER
2064 * INIT CONTAINS CODE FOR INITIALIZATION
2065 * DONE FIRST TIME PXRDT EXECUTED ONLY
2066 *
2067 00A0 PGEND EQU >A0 START OF BUFFER AREA
2068 * REF ENDBUF END OF BUFFER
2069 *
2070 INIT
2071 *
2072 * INITIALIZE XOP VECTOR
2073 REF SVCWP
2074 REF SVCSR
2075 1040 0209 LI R9, SVCWP
1042 0000
2076 1044 C809 MOV R9, @>7C
1046 007C
0077 1048 0209 LI R9, SVCSR
104A 0000
2078 104C C809 MOV R9, @>7E
104E 007E
2079 *
2080 * PRINT OUT HEADING ETC.
2081 *
2082 1050 C2A0 MOV @KBLUND, R10 OPEN LOG KEYBOARD
1052 019A'
2083 1054 0420 BLWP @OPEN
1056 0F68'
2084 1058 C2A0 MOV @PRLUND, R10 OPEN LOG PRINTER
105A 019C'
2085 105C 0420 BLWP @OPEN
105E 0F68'
2086 1060 020A LI R10, LFCR PRINT LINE FEED/CARRIAGE RETU
1062 0184'
2087 1064 0420 BLWP @PRINT
1066 0FEC'
2088 1068 020A LI R10, NAMMSG PRINT PROGRAM TITLE
106A 1072'
2089 106C 0420 BLWP @PRINT
106E 0FEC'
2090 *
2091 * RETURN TO MAIN PROGRAM
2092 1070 045B RT

```

PXREDT
INITIALIZATION

MIRA990

V2L1 12:13:11
948927-9901**

258/76

PAGE 0066

0994
0995
2096

*
*
*

MESSAGES

2097 1072 0A
2098 1073 50
2099 108C 0A
108D 0D

NAMMSG BYTE LFE
TEXT 'PXREDT 948927 ** 12JUL76'
BYTE LFE, CRE

2100

END START

ABNMS	0612	R	ABRLFG	011C	ACTION	0002	R	ANCHOR	017E		
ATTOP	03D6	R	B DPRNT	0CF4	BELL	0007	R	BINASC	0194		
R	BLD1	0C84	R	BLD1A	0C9A	R	BLD2	0CAA	R	BLD3	0CB2
R	BOTTOM	065A	R	BTM	0640	R	BTM1	065E		BUFADR	0005
	BUFLEN	0006	R	C1LUNO	019E	R	C2LUNO	01A0	R	CC	0138
R	CHFLG	0116	R	CHNG	066E		CHRCNT	0007	R	CHRET	0EB8
	CLINNO	0008		CLLOC	0006	R	CLRDUM	01E0	R	CLRLIN	0E8E
R	CMDLST	02A0	R	CMPR	024A	R	CNVRT	0008	R	COMMND	0226
R	CONT	0196	R	CONVRT	0D04	R	CPBUF	006E	R	CPBUFA	006C
R	CR	0185		CRE	000D	R	CURNO	0150	R	CURPTL	060A
R	DFLT	0B38		DUMNXT	0000	R	DWN	0680	R	DWN1	068A
R	DWN2	06A8	R	DWN3	069A	R	DWN4	06AE	D	EDTCSR	01A6
R	ENDACT	027A	E	ENDBUF	0008	R	ENDFIL	0F84	R	ENDLST	02AE
R	ENDPRG	01A2		ENTRY	0000		EOF	2000	R	EOFCOD	01A5
R	EOFEXT	06B4	R	EOFLG	0114	R	EPEXIT	029C	R	EREXFN	0A58
R	EREXMV	0C34	R	EREXOP	063C	R	EREXST	0F3A	R	ESCRTN	021C
R	EX1	05FA	R	EXEXOP	063A	R	EXIT	06BA	R	EXITIN	0792
R	EXOP	0590	R	EXOPEX	05F4	R	EXT	0794	R	EXTBDP	0D36
R	EXTBP1	0D4A	R	EXTTEXT	057E	R	EXTFLG	0118	R	EXTINS	0AFC
R	EXTMV	0C32	R	EXTPRT	0C66	R	EXTQT	0DC8	R	EXTRMV	0EB0
R	EXTSCR	0E2A	R	EXTSK	0518	R	EXTSTP	0F38	R	EXTUP	06FA
R	FASK	084A		FCHAR	0008	R	FDXT	04FE	R	FERXIT	0A3A
R	FFLG	0128		FGPOS	000D	R	FINLAD	014C		FLGS	0004
R	FLIN	04E4	R	FLIN1	04E8	R	FND	0798	R	FND1	0874
R	FND10	09CA	R	FND11	09EC	R	FND11A	09FC	R	FND11B	09F2
R	FND11C	0A20	R	FND1A	089A	R	FND1B	086E	R	FND2	08CE
R	FND3	08E4	R	FND4	08D8	R	FND5	0900	R	FND6	0908
R	FND7	08F8	R	FND8	0968	R	FND9	0978	R	FND9A	098E
R	FND9B	0984	R	FNDA	07C2	R	FNDB	07E4	R	FNDC	0800
R	FNDF	082A		FPPOS	000E	R	FROLIN	0994	R	FRSTR	0986
R	FSD	0164	R	FSDX	0174	R	FSDXT	0172	R	FXLAD	016A
R	GCEXT	103C	R	GCHA	0580	R	GLOOP	102E	R	GETANS	02DA
R	GETCH	052E	R	GETCH1	055A	R	GETCNT	102C	R	GETCOM	0232
R	GITCR	0A5C	R	GITCR1	0A5E	R	GITCRB	0A6A	R	GULM	061E
R	HICOR	0008	R	HIMRG	0136	R	INIT	1040	R	INLNNO	017A
R	INPS1	078E	R	INPS2	0752	R	INPS3	0722	R	INSCH	0A80
R	INSOU	0708	R	INSR1	0A9A	R	INSR1A	0A9C	R	INSR5	0A8A
R	INSR6	0AEA	R	INSR7	0AF6	R	INSRT	0A7C	R	INTLPT	0180
	IOC	0002		IOERR	4000		IOPLUN	0003	R	IOWKS	004C
R	KBLUNO	019A	R	KEEP1	0B24		KEY	0006	R	KEYIN	0FA4
R	KEYPLF	0FD2	R	KEYRD	0FAC	R	KLRLIN	0E16	R	KP	0B0A
R	KP2	0B14	R	LAMOV	012C		LEN	000A	R	LF	0184
R	LFCR	0184		LFE	000A	R	LFFLG	011E		LINAD	0002
R	LINADD	0144	R	LINADR	0146		LINPTR	0004	R	LLIMIT	0182
R	LML	015A	R	LMTS	0B44	R	LOMRG	0134	R	LSTLIN	03CC
R	MAINRG	0006	R	MALIN	012A	R	MARGLM	012E	R	MAXLIN	015C
R	MCOUNT	02E4	R	MDFLT	0F3E	R	MEMPTY	03EA	R	MOEXT	0B3E
R	MSG00	02FC	R	MSG01	0317	R	MSG02	032C	R	MSG03	033C
R	MSG04	034E	R	MSG05	035B	R	MSG06	036C	R	MSG07	0379
R	MSG08	0380	R	MSG09	038B	R	MSG10	0398	R	MSG11	03A2
R	MSG12	03B0	R	MSG13	03B4	R	MSG14	03BA	R	MSGOUT	02CE
R	MV	0B88	R	MV1	0BB8	R	MV4	0BD6	R	MV5	0BF4
R	MV6	0BFA	R	MV6A	0C0A	R	MV6B	0C2C	R	MV7	0BE8
R	MVRCNT	0178	R	NAMMSG	1072	R	NCH	0538		NEWPTR	000F
R	NMEXIT	0516	R	NOTFND	03F4	R	NOVY	0958	R	NULREL	05D0

O NUMB	013C	R NUMBF	013A	R NUMEXT	0526	R NULF	0126
NXIT	055E	R NXLAD	014A	R NXTLAD	014E	R OKXT	0574
R OPEN	0F68	R OPERR	03FE	R OFNCOD	01A4	R OPNERR	03E0
R PABS	0C5E	R PBUF	00C2	R PBUFA	00C0	R PFKNT	0A40
R PFLG	0120	PGEND	00A0	R PLFLG	011A	PLIMTS	0003
R PMARGIN	0132	R PRELAD	0148	R PRENT	0FFA	R PREXT	102A
R PRINT	0FEC	R PRINTN	0FF0	R PRLAD	0152	R PRLIN	0C68
R PRLUND	019C	R PRNT	0C38	R PRNT1	0C42	R PRNT2	0C52
PROMPT	0003	PRT	0004	R PRTB	0C0C	R PRTB1	0CCE
R PRTCR	0FF8	R PRTKEY	1008	PRTN	0005	R PRTNCR	0FF4
R PRTPOS	0188	R PRTSP	0CDE	R PRVLIN	015E	R PRWRT	0FFE
R PTONLN	09A2	R PTSP1	0D2C	R PUTSPS	0D28	R PXREDT	01AE
R QT	0D4E	R QT0	0D66	R QT1	0D76	R QT2	0D7A
R QT3	0D94	R QT4	0DC0	R QT5	0DA6	R0	0000
R1	0001	R10	000A	R11	000B	R12	000C
R13	000D	R14	000E	R15	000F	R2	0002
R3	0003	R4	0004	R5	0005	R6	0006
R7	0007	R8	0008	R9	0009	RDCOD	0900
RDCODP	0000	R READ	0F9A	R RELNEG	05E4	R RELNUM	05BE
R RELRMV	0EA2	R REPFLG	0124	R REXB	0EA6	R RLNEG1	05E6
R RMV	0E34	R RMV1	0E4E	R RMV2	0E86	R RMV4	0E7A
R RMV6	0E3C	R RMVCH	0E38	R ROUTAD	02B0	R ROUTR1	002C
R ROUTRG	0026	RTN	000B	R SAVALL	0DB4	R SAVMRG	0162
R SAVRTN	0154	R SCBLK	0050	R SCFP	016E	R SCR0	0DEC
R SCR1	0E0A	R SCR2	0E12	R SCR3	0E2C	R SCR4	0E28
R SCROLL	0DCE	R SFND	0934	R SKANOP	0500	R SKANTM	051A
SLADDR	013E	R SLIN	0EEC	R SLINPT	0158	R SLOC	0156
SMARG	0F12	R SNLIN	0EE2	R SF	0186	R SPCR	0186
R SPLIM	0EF6	R SQ1	0410	R SQ2	041C	R SQ3	0472
R SQ3A	0476	R SQ4	043C	R SQ4A	0460	R SQ5	04A0
R SQ7	04C0	R SQ7A	04E2	R SQ8	0420	R SQ9	047C
R SQNLAD	0142	R SQPLAD	0140	R SQUEEZ	0408	R SRKNT	0166
R SSD	0168	R SSDX	0176	R START	01A6	R STP	0EBE
R STRCHR	0170	R SUCCES	025A	X SVC	000F	E SVCSR	104A
E SVCWP	1042	R SVPRM	0160	R TAB	0198	R TAB5	018A
R TEN	017C	R TERM	0197	R THLAD	016C	TMLOC	0007
R TONLIN	09A8	R TOP	0F4C	TOPMEM	0001	R TP1	0F5E
UDCNT	0001	R UDCOM1	06BC	R UP	06CA	R UP1	06EA
R UP2	06FC	R UP3	06F6	R UP4	06E4	R VFLG	0122
R VLNPTR	002E	WRCOD	0B00	WRCODP	0001	R WRITE	0FE0
R YEA	0195						

0000 ERRORS


```
0002          IDT  'PXRBUF'
0003      * TITLE:  PXRBUF
0004      *          PX990 RESIDENT MONITOR BUFFER DEFINITION
0005      * REVISION:
0006      *          ORIGINAL
0007      * COMPUTER: 990, 990 ASSEMBLY
0008      * ABSTRACT: PXRBUF DEFINES THE END OF A BUFFER AREA EXTENDIN
0009      *          FROM >A0 TO THE BEGINNING OF THE ASSEMBLER, TEXT
0010      *          EDITOR, MONITOR.
0011      * CALLING SEQUENCE: NON-CALLABLE
0012 0000          RORG
0013      *
0014      * DEFINE END OF ASSEMBLER SYMBOL TABLE.
0015      *
0016          DEF  BGNPXR
0017 0000  BGNPXR BSS  1
0018          DEF  SYMT
0019 0001  SYMT   BSS  1          THE LAST BYTE OF THE BUFFER AREA.
0020      *
0021      * DEFINE END OF BUFFER FOR TEXT EDITOR AND LINKING LOADER
0022      *
0023          DEF  ENDBUF
0024 0002' ENDBUF EQU  $
0025          END
```

NO ERRORS

948928-9901

**

NAME	VALUE	ADDR	REFERENCES
\$	R	0002	0024
BGNPXR	D	0000	0017 0016
BSS			0017 0019
DEF			0016 0018 0023
END			0025
ENDBUF	D	0002	0024 0023
EQU			0024
IDT			0002
RORG			0012
SYMT	D	0001	0019 0018
TITL			0001


```

0004          IDT  'CMDEFQ'
0005          * TITLE:  CMDEFQ
0006          *          COMMAND DEFINITION TABLE
0007          * REVISION: 2/1/76
0008          *          ORIGINAL
0009          *          07/15/76
0010          *          EXTENDED TO INCLUDE PA AND TE COMMANDS
0011          * REVISION: 8/30/76
0012          *          MODIFIED FOR A PERMANENT TRACE MODULE (FORMERLY
0013          *          OVERLAY)
0014          * COMPUTER: 990,ASM
0015          * ABSTRACT: CONTAINS A LIST OF THE COMMANDS, THEIR
0016          *          ALLOWABLE PARAMETERS, AND THE ENTRY POINT
0017          *          FOR THE COMMAND PROCESSER ROUTINE.
0018          * CALLING SEQUENCE:
0019          *          NON EXECUTABLE
0020          *
0021          *          COMMAND STRUCTURE
0022          *
0023          *          *****
0024          *          *          COMMAND          *
0025          *          *****
0026          *          *          P1 - P8          *
0027          *          *****
0028          *          *          SERVICE ROUTINE *
0029          *          *****
0030          *
0031          *          PARAMETER DEFINITIONS
0032          *
0033          0001 PH      EQU      1          HEX STRING
0034          0002 PC      EQU      2          CHARACTER STRING
0035          0003 PN      EQU      3          NULL
0036          *

```

```

0038      *
0039      *      DEF'S AND REF'S
0040      *
0041      *      DEF  CMDTBL
0042      *      DEF  NOCMDS
0043      *      DEF  SREGN
0044      *
0045      *
0046      *      0000' CMDTBL EQU  $
0047      *
0048      *      ASSIGN LUN0
0049      *
0050      *      REF  ASLUN
0051      *      TEXT 'AL'
0051 0000  41
0051 0001  4C
0052 0002 6FFF      DATA PH*4+PC*4+PN*4+PN*4+PN*4+PN*4+PN
0053 0004 0000      DATA ASLUN
0054      *
0055      *      DUMP IN ABSOLUTE FORMAT
0056      *
0057 0006  44      TEXT 'DP'
0057 0007  50
0058 0008 56BF      DATA PH*4+PH*4+PH*4+PC*4+PC*4+PN*4+PN*4+PN
0059 000A 0000      DATA 0
0060      *
0061      *      DUMP IN BNPf FORMAT
0062      *
0063 000C  44      TEXT 'DB'
0063 000D  42
0064 000E 97FF      DATA PC*4+PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN
0065 0010 0000      DATA 0
0066      *
0067      *      DUMP IN HIGH/LOW FORMAT
0068      *
0069 0012  48      TEXT 'HL'
0069 0013  4C
0070 0014 95FF      DATA PC*4+PH*4+PH*4+PH*4+PN*4+PN*4+PN*4+PN
0071 0016 0000      DATA 0
0072      *
0073      *      EXECUTE USER PROGRAM
0074      *
0075      *      REF  EXCT
0076 0018  45      TEXT 'EX'
0076 0019  58
0077 001A FFFF      DATA PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0078 001C 0000      DATA EXCT
0079      *
0080      *      LINK AND LOAD
0081      *
0082      *      REF  LALCSR
0083 001E  4C      TEXT 'LL'
0083 001F  4C
0084 0020 FFFF      DATA PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0085 0022 0000      DATA LALCSR
0086      *
0087      *      STANDARD LOADER
0088      *
0089      *      REF  LOAD
0090 0024  4C      TEXT 'LP'
0090 0025  50
    
```

```
0091 0026 5FFF      DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0092 0028 0000      DATA LOAD
0093                *
0094                *   ABSOLUTE LOADER
0095                *
0096 002A  4C       TEXT 'LA'
      002B  41
0097 002C 7FFF      DATA PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0098 002E 0000      DATA 0
0099                *
0100                *
0101                *   LOAD UP FRONT LOADER AND PROGRAM
0102                *
0103                REF LDUFL
0104 0030  4C       TEXT 'LU'
      0031  55
0105 0032 5FFF      DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0106 0034 0000      DATA LDUFL
0107                *
0108                *   LOAD FROM PROGRAMMER
0109                *
0110                REF PL
0111 0036  50       TEXT 'PL'
      0037  4C
0112 0038 5FFF      DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0113 003A 0000      DATA PL
0114                *
0115                *   OVERLAY SUPERVISOR
0116                *
0117                REF OVLY
0118 003C  4F       TEXT 'OV'
      003D  56
0119 003E 7FFF      DATA PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0120 0040 0000      DATA OVLY
0121                *
0122                *   PROM PROGRAMMER
0123                *
0124 0042  50       TEXT 'PP'
      0043  50
0125 0044 9555      DATA PC*4+PH*4+PH*4+PH*4+PH*4+PH*4+PH*4+PH
0126 0046 0000      DATA 0
0127                *
0128                *   STANDARD PROM PROGRAMMER
0129                *
0130 0048  50       TEXT 'PS'
      0049  53
0131 004A AFFF      DATA PC*4+PC*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0132 004C 0000      DATA 0
0133                *
0134                *   RUN
0135                *
0136                REF RUN
0137 004E  52       TEXT 'RU'
      004F  55
0138 0050 7FFF      DATA PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0139 0052 0000      DATA RUN
0140                *
0141                *   INSPECT CRU
0142                *
0143                REF ICP
```

```
0144 0054 49 TEXT 'IC'
      0055 43
0145 0056 5FFF DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0146 0058 0000 DATA ICP
0147 *
0148 * INSPECT MEMORY
0149 *
0150 REF IMP
0151 005A 49 TEXT 'IM'
      005B 4D
0152 005C 5FFF DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0153 005E 0000 DATA IMP
0154 *
0155 * INSPECT REGISTERS
0156 *
0157 REF IRP
0158 0060 49 TEXT 'IR'
      0061 52
0159 0062 FFFF DATA PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0160 0064 0000 DATA IRP
0161 *
0162 * INSPECT SNAPSHOT
0163 *
0164 REF ISS
0165 0066 49 TEXT 'IS'
      0067 53
0166 0068 5FFF DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0167 006A 0000 DATA ISS
0168 *
0169 * INSPECT WORKSPACE
0170 *
0171 REF IWP
0172 006C 49 TEXT 'IW'
      006D 57
0173 006E 5FFF DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0174 0070 0000 DATA IWP
0175 *
0176 * MODIFY CRU
0177 *
0178 REF MODCRU
0179 0072 4D TEXT 'MC'
      0073 43
0180 0074 5FFF DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0181 0076 0000 DATA MODCRU
0182 *
0183 * MODIFY MEMORY
0184 *
0185 REF MODMEM
0186 0078 4D TEXT 'MM'
      0079 4D
0187 007A 7FFF DATA PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0188 007C 0000 DATA MODMEM
0189 *
0190 * MODIFY REGISTERS
0191 *
0192 REF MRP
0193 007E 4D TEXT 'MR'
      007F 52
0194 0080 FFFF DATA PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0195 0082 0000 DATA MRP
```

```

0196      *
0197      *      MODIFY WORKSPACE REGISTERS
0198      *
0199      REF  MODWSP
0200 0084 4D      TEXT 'MW'
      0085 57
0201 0086 7FFF      DATA PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0202 0088 0000      DATA MODWSP
0203      *
0204      *
0205      *      SET BREAKPOINT
0206      *
0207      REF  SBP
0208 008A 53      TEXT 'SB'
      008B 42
0209 008C 55FF      DATA PH*4+PH*4+PH*4+PH*4+PN*4+PN*4+PN*4+PN
0210 008E 0000      DATA SBP
0211      *
0212      *      SET SNAPSHOT
0213      *
0214      REF  SSS
0215 0090 53      TEXT 'SS'
      0091 53
0216 0092 557F      DATA PH*4+PH*4+PH*4+PH*4+PH*4+PN*4+PN*4+PN
0217 0094 0000      DATA SSS
0218      *
0219      *      SET REGION
0220      *
0221      REF  SR
0222 0096 53      TEXT 'SR'
      0097 52
0223 0098 5595      DATA PH*4+PH*4+PH*4+PH*4+PC*4+PH*4+PH*4+PH
0224 009A 0000 SREGN DATA SR
0225      *
0226      *      SET TRACE
0227      *
0228      REF  STRACE
0229 009C 53      TEXT 'ST'
      009D 54
0230 009E 6FFF      DATA PH*4+PC*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0231 00A0 0000      DATA STRACE
0232      *
0233      *      SET PROTECT REGION
0234      *
0235      REF  LDPRT
0236 00A2 53      TEXT 'SP'
      00A3 50
0237 00A4 5FFF      DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0238 00A6 0000      DATA LDPRT
0239      *
0240      *      CLEAR BREAKPOINT
0241      *
0242      REF  CBP
0243 00A8 43      TEXT 'CB'
      00A9 42
0244 00AA 5FFF      DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0245 00AC 0000      DATA CBP
0246      *
0247      *      CLEAR SNAPSHOTS
0248      *

```


0249			REF CSS
0250	00AE	43	TEXT 'CS'
	00AF	53	
0251	00B0	5FFF	DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0252	00B2	0000	DATA CSS
0253			*
0254			* CLEAR REGION
0255			*
0256			REF CRR
0257	00B4	43	TEXT 'CR'
	00B5	52	
0258	00B6	5FFF	DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0259	00B8	0000	DATA CRR
0260			*
0261			* CLEAR PROTECT REGION
0262			*
0263			REF CLEARP
0264	00BA	43	TEXT 'CP'
	00BB	50	
0265	00BC	FFFF	DATA PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0266	00BE	0000	DATA CLEARP
0267			*
0268			* FIND WORD
0269			*
0270			REF FWD
0271	00C0	46	TEXT 'FW'
	00C1	57	
0272	00C2	55FF	DATA PH*4+PH*4+PH*4+PH*4+PN*4+PN*4+PN*4+PN
0273	00C4	0000	DATA FWD
0274			*
0275			* FIND BYTE
0276			*
0277			REF FBT
0278	00C6	46	TEXT 'FB'
	00C7	42	
0279	00C8	55FF	DATA PH*4+PH*4+PH*4+PH*4+PN*4+PN*4+PN*4+PN
0280	00CA	0000	DATA FBT
0281			*
0282			* HEX ARITHMETIC
0283			*
0284			REF HXAR
0285	00CC	48	TEXT 'HA'
	00CD	41	
0286	00CE	5FFF	DATA PH*4+PH*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0287	00D0	0000	DATA HXAR
0288			*
0289			* ONE PASS ASSEMBLER
0290			*
0291	00D2	50	TEXT 'PA'
	00D3	41	
0292	00D4	FFFF	DATA PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0293			REF ASMCSR
0294	00D6	0000	DATA ASMCSR
0295			*
0296			* TEXT EDITOR
0297			*
0298	00D8	54	TEXT 'TE'
	00D9	45	
0299	00DA	FFFF	DATA PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN*4+PN
0300			REF EDTCSR

0301 00DC 0000 DATA EDTCSR

0302 *

0303 *

0304 *

0305 00DE' NOCMDS EQU \$

0306 00DE 0025 DATA 37

0307 END

NO ERRORS

\$	R	00E0		0046	0305									
ASLUN	E	0004	0050	0053										
ASMCSR	E	00D6	0293	0294										
CBP	E	00AC	0242	0245										
CLEARP	E	00BE	0263	0266										
CMDTBL	D	0000	0046	0041										
CRR	E	00B8	0256	0259										
CSS	E	00B2	0249	0252										
DATA				0052	0053	0058	0059	0064	0065	0070	0071	0077		
				0078	0084	0085	0091	0092	0097	0098	0105	0106		
				0112	0113	0119	0120	0125	0126	0131	0132	0138		
				0139	0145	0146	0152	0153	0159	0160	0166	0167		
				0173	0174	0180	0181	0187	0188	0194	0195	0201		
				0202	0209	0210	0216	0217	0223	0224	0230	0231		
				0237	0238	0244	0245	0251	0252	0258	0259	0265		
				0266	0272	0273	0279	0280	0286	0287	0292	0294		
				0299	0301	0306								
				0041	0042	0043								
DEF														
EDTCSR	E	00DC	0300	0301										
END				0307										
EQU				0033	0034	0035	0046	0305						
EXCT	E	001C	0075	0078										
FBT	E	00CA	0277	0280										
FWD	E	00C4	0270	0273										
HXAR	E	00D0	0284	0287										
ICP	E	0058	0143	0146										
IDT				0004										
IMP	E	005E	0150	0153										
IRP	E	0064	0157	0160										
ISS	E	006A	0164	0167										
WP	E	0070	0171	0174										
LALCSR	E	0022	0082	0085										
LDPRT	E	00A6	0235	0238										
LDUFL	E	0034	0103	0106										
LOAD	E	0028	0089	0092										
MODCRU	E	0076	0178	0181										
MODMEM	E	007C	0185	0188										
MODWSP	E	0088	0199	0202										
MRP	E	0082	0192	0195										
NOCMDS	D	00DE	0305	0042										
OVLY	E	0040	0117	0120										
PAGE				0002	0003	0037								
PC		0002	0034	0052	0058	0058	0064	0070	0125	0131	0131	0223		
				0230										
PH		0001	0033	0052	0058	0058	0058	0064	0064	0070	0070	0070		
				0091	0091	0097	0105	0105	0112	0112	0119	0125		
				0125	0125	0125	0125	0125	0125	0138	0145	0145		
				0152	0152	0166	0166	0173	0173	0180	0180	0187		
				0201	0209	0209	0209	0209	0216	0216	0216	0216		
				0216	0223	0223	0223	0223	0223	0223	0223	0230		
				0237	0237	0244	0244	0251	0251	0258	0258	0272		
				0272	0272	0272	0279	0279	0279	0279	0286	0286		
PL														
PN	E	003A	0110	0113										
		0003	0035	0052	0052	0052	0052	0052	0052	0058	0058	0058		
				0064	0064	0064	0064	0064	0070	0070	0070	0070		
				0077	0077	0077	0077	0077	0077	0077	0077	0084		
				0084	0084	0084	0084	0084	0084	0084	0084	0091	0091	
				0091	0091	0091	0091	0097	0097	0097	0097	0097		
				0097	0097	0105	0105	0105	0105	0105	0105	0112		
				0112	0112	0112	0112	0112	0119	0119	0119	0119		

0119	0119	0119	0131	0131	0131	0131	0131	0131	0131
0138	0138	0138	0138	0138	0138	0138	0138	0145	0145
0145	0145	0145	0145	0152	0152	0152	0152	0152	0152
0152	0159	0159	0159	0159	0159	0159	0159	0159	0159
0166	0166	0166	0166	0166	0166	0166	0173	0173	0173
0173	0173	0173	0180	0180	0180	0180	0180	0180	0180
0187	0187	0187	0187	0187	0187	0187	0187	0194	0194
0194	0194	0194	0194	0194	0194	0194	0201	0201	0201
0201	0201	0201	0201	0209	0209	0209	0209	0209	0216
0216	0216	0230	0230	0230	0230	0230	0230	0230	0237
0237	0237	0237	0237	0237	0244	0244	0244	0244	0244
0244	0244	0251	0251	0251	0251	0251	0251	0251	0258
0258	0258	0258	0258	0258	0265	0265	0265	0265	0265
0265	0265	0265	0265	0272	0272	0272	0272	0272	0279
0279	0279	0279	0286	0286	0286	0286	0286	0286	0286
0292	0292	0292	0292	0292	0292	0292	0292	0292	0299
0299	0299	0299	0299	0299	0299	0299	0299	0299	0299
0050	0075	0082	0089	0103	0110	0117	0136	0143	0143
0150	0157	0164	0171	0178	0185	0192	0199	0207	0207
0214	0221	0228	0235	0242	0249	0256	0263	0270	0270
0277	0284	0293	0300						

REF

RUN E 0052 0136 0139
 SBP E 008E 0207 0210
 SR E 009A 0221 0224
 SREGN D 009A 0224 0043
 SSS E 0094 0214 0217
 STRACE E 00A0 0228 0231

TEXT 0051 0057 0063 0069 0076 0083 0090 0096 0104
 0111 0118 0124 0130 0137 0144 0151 0158 0165
 0172 0179 0186 0193 0200 0208 0215 0222 0229
 0236 0243 0250 0257 0264 0271 0278 0285 0291
 0298
 TITL 0001


```

0005          IDT 'PRLDDR'
0006 * TITLE:   PR990 LOADER DRIVER
0007 *          PRLDDR
0008 * REVISION:
0009 *          ORIGINAL
0010 * COMPUTER: 990, ASM
0011 * ABSTRACT: PRLDDR IS RESPONSIBLE FOR FOUR FUNCTIONS:
0012 *           1) PROCESS LP, LA KEYBOARD COMMANDS AND
0013 *              CALL ROM LOADER AND ABS LOADER RESPECTIVEL
0014 *           2) INTERFACE TO THE ROM LOADER
0015 *           3) PERFORM IO TO SUPPORT THE ROM LOADER.
0016 *           4) CALL ROM LOADER TO LOAD OVERLAY
0017 * CALLING SEQUENCE:
0018 *           1- KEYBOARD COMMAND
0019 *              R10 - POINTER TO COMMAND PARAMETER LIST
0020 *              BL @LOAD
0021 *              LOAD PROGRAM
0022 *              BL @LDABS
0023 *              LOAD ABSOLUTE
0024 *              BL @LOADOV
0025 *              LOAD OVERLAY
0026 *           ENTRY
0027 *              R10 - POINTER TO COMMAND PARM LIST
0028 *              R1  - LOAD BIAS FOR OV COMMAND ONLY
0029 *           EXIT
0030 *              R0 NOT EQUAL TO 0 FOR ERROR
0031 *           2- LOADER IO
0032 *              BL @R11          (MAIN DRIVER SETS POINTER
0033 *                               IN R11)
0034 * PROCEDURE LOADER_DRIVER(CPL);
0035 * /* LOADER_DRIVER WILL ACCEPT THE
0036 *    INPUT LUNO AND CALL THE LOADER
0037 *    TO LOAD THE USER'S PROGRAM.
0038 * /*
0039 * LOADER_LUNO = 7; /* DEF = CS1 */
0040 * IF CPL.PARM1 .NE. NULL THEN
0041 *   LOADER_LUNO = CPL.PARM1
0042 * LOAD_PRB_LUNO = LOADER_LUNO;
0043 * IF DEVICE TYPE .NE. CS SIGNAL ERROR
0044 * LOAD_PRB_OPCODE = 'OPEN';
0045 * CALL SVC(SVC_CALL_BLOCK);
0046 * IF<LD FLAG.LE. 1> THEN
0047 *   IF<CPL_PARM2 .NE. NULL>THN
0048 *     BIAS= PARM2;
0049 *   END IF;
0050 *   ELSE BIAS=DEFAULT_BIAS;
0051 * CALL SET LOADER WORKSPACE;
0052 * CALL SET_WORKSPACE(R11, LOC(LOADER_IO));
0053 * CALL SETWORKSPACE(R9, BIAS)
0054 * CALL LOADER(LOADER_IO,
0055 *   BIAS, CASS_FLAG)
0056 * IF ERROR .NE. 0 THEN DO;
0057 *   IF PROGRAM_NAME_CHAR_COUNT .NE. 0
0058 *     THEN CALL PRINT<PROGRAM_NAME>;
0059 *   IF PROGRAM_ENTRY .NE. 0 THEN
0060 *     IF CALLER NE OVLY THEN
0061 *       USER_PC = PROGRAM_ENTRY
0062 *     IF CALLER EQ LOADUPFRONT THEN
0063 *       CALL UPFRONT LOADER
0064 * END; ELSE

```



```

0065 *          CALL ERROR('LD00');
0066 *          END;
0067 *          RETURN;
0068 *          PROCEDURE LOADER_IO;
0069 *          /* LOADER_IO PROVIDES IO SUPPORT
0070 *             FOR THE ROM LOADER.  THE OPEN
0071 *             HAS BEEN PROCESSED BY THE MAIN
0072 *             DRIVER.
0073 *          */
0074 *             CHAR_COUNT=CHAR_COUNT -1;
0075 *             IF CHAR_COUNT =-1 THEN DO;
0076 *                 LOAD_PRB_OPCODE = 'READ';
0077 *                 CALL SVC(SVC_CALL_BLOCK);
0078 *                 IF ERROR THEN EXIT;
0079 *                 IF EOF THEN EXIT;
0080 *                 CURR_BUFF_ADDR = LOADER_BUFFER_ADDR;
0081 *                 OUTPUT LOAD ADDRESS TO LIGHTS
0082 *             END;
0083 *             CALL MOVE_BIT(RETURN_VALUE, CURR_BUF_ADDR);
0084 *             CURR_BUFF_ADDR = CURR_BUFF_ADDR+1;
0085 *          END;
0086 *          END LDIO;
0087 *          DO FOREVER;
0088 *              CALL GETCHAR;
0089 *          END
0090 *          END LOADER_DRIVER;

```

REF'S AND DEF'S

```

0094 DEF  LDUFL
0095 DEF  LDABS
0096 DEF  LOAD
0097 DEF  LOADOV
0098 REF  SVCALT
0099 REF  SVCWP
0100 REF  SVCSR
0101 REF  SAYWP
0102 REF  CS
0103 REF  CLDT
0104 REF  OVLRET
0105 REF  LDPRB
0106 REF  LDOPCD
0107 REF  LDLUN
0108 REF  LDFLG
0109 REF  LDADDR
0110 REF  LDCC
0111 REF  LDCBA
0112 REF  LDBUF
0113 REF  CR
0114 REF  ABSLDR
0115 REF  ROMLDR
0116 REF  LDTBL
0117 REF  PRNTC
0118 REF  PRCLF
0119 REF  TRNARA
0120 REF  USRPC
0121 REF  ERROR
0122 REF  LDCBBC
0123 REF  LDNMCC
0124 REF  DFBIAS

```

```

0125 REF INIT
0126 REF CRUOFF
0127 REF GETBUF
0128 REF RETBUF
0129 REF LWP
0130 REF RWP
0131 REF ACL
0132 REF RR
0133 REF BGNPXR PXRMAE FIRST ADDRESS
    
```

```

*
0134 *
0135 *WORKSPACE REGISTER DEFINITIONS
0136 *
    
```

```

0137 0000 R0 EQU 0
0138 0001 R1 EQU 1
0139 0002 R2 EQU 2
0140 0003 R3 EQU 3
0141 0004 R4 EQU 4
0142 0005 R5 EQU 5
0143 0006 R6 EQU 6
0144 0007 R7 EQU 7
0145 0008 R8 EQU 8
0146 0009 R9 EQU 9
0147 000A R10 EQU 10
0148 000B R11 EQU 11
0149 000C R12 EQU 12
0150 000D R13 EQU 13
0151 000E R14 EQU 14
0152 000F R15 EQU 15
    
```

```

0153 *
0154 DXOP SVC, 15
0155 0100 LBYTE EQU 256
0156 0000 OPN EQU 0
0157 0009 READ EQU 9
0158 2400 LDERR EQU >2400
0159 2401 LUNERR EQU >2401 INVALID LUN0
    
```

```

0160 *
0161 * UPFRONT LOADER DEFAULT BIAS
0162 *
0163 FE50 UFLBIA EQU ->1B0
0164 *
    
```

```

0165 * 1-PROCEDURE LOADER_DRIVER(CPL)
0166 * 2-/* LOADER_DRIVER WILL ACCEPT
0167 * 2- INPUT LUN0 AND CALL THE L
0168 * 2- TO LOAD THE USER'S PROGRA
0169 * 2-*/
0170 *
    
```

```

0171 ***** UP FRONT LOADER ENTRY POINT *****
    
```

```

0172 0000' LDUFL EQU $
0173 0000 04CE CLR R14
0174 0002 1005 JMP LOA002
    
```

```

0175 ***** ROM LOADER ENTRY POINT *****
    
```

```

0176 0004' LOAD EQU $
0177 0004 020E LI R14, 1
0178 0006 0001 JMP LOA002
0179 0008 1002
    
```

```

0179 ***** ABS LOADER ENTRY POINT *****
    
```

```

0180 000A' LDABS EQU $
0181 000A 020E LI R14, 2
0182 000C 0002
    
```

```

0182 *
    
```

```

0183          *      XOP INTERRUPT VECTOR
0184      000E'  LOA002 EQU $
0185      000E 0209      LI   R9, SVCWP
          0010 0000
0186      0012 C809      MOV   R9, @>7C
          0014 007C
0187      0016 0209      LI   R9, SVCSR
          0018 0000
0188      001A C809      MOV   R9, @>7E
          001C 007E

0189          *
0190      001E 1002      JMP   LOA005
0191          ***** OVERLAY ENTRY POINT *****
0192      0020'  LOADOV EQU $
0193      0020 020E      LI   R14, 3
          0022 0003
0194      0024'  LOA005 EQU $
0195          *      *ALLOC, COPY, LINK
0196      0024 0420      BLWP @ACL
          0026 0000
0197      0028 C80D      MOV   R13, @SAVWP
          002A 0000
0198      002C C0ED      MOV   @R14*2(R13), R3      RETRIEVE R14 FROM OLD WS
          002E 001C
0199      0030 C10A      MOV   R10, R4
0200          *
0201      0032 0209      LI   R9, 7*LBYTE      2-LOADER_LUNO = 7; /* DEF = C
          0034 0700

0202          *
0203          *
0204      0036 C034      MOV   *R4+, R0
0205      0038 0A90      SLA  R0, 9      CHECK PRESENCE BITS
0206      003A 1702      JNC  LOA010
0207      003C C274      MOV   *R4+, R9
0208      003E 0A89      SLA  R9, 8      BYTE SIZE PARM
0209          0040'  LOA010 EQU $
0210          *
0211      0040 D809      MOVB R9, @LDLUN      2-LOAD_PRB_LUNO = LOADER_LUNO;
          0042 0000

0212          *
0213      0044 C289      MOV   R9, R10      2-IF DEVICE TYPE .NE. CS SIGNA
0214      0046 098A      SRL  R10, 8
0215      0048 06A0      BL   @CLDT
          004A 0000
0216      004C 0208      LI   R8, CS
          004E 0000
0217      0050 0989      SRL  R9, 8
0218      0052 020A      LI   R10, LUNERR
          0054 2401
0219      0056 8209      C    R9, R8
0220      0058 164C      JNE  LOA052
0221          *
0222      005A 0209      LI   R9, OPN*LBYTE      2-LOAD_PRB_OPCODE = 'OPEN';
          005C 0000
0223      005E D809      MOVB R9, @LDOPCD
          0060 0000

0224          *
0225      0062 020A      LI   R10, LDPRB      2-CALL SVC(SVC_CALL_BLOCK);
          0064 0000
0226      0066 0420      BLWP @SVCALT
    
```

```

0068 0000
0227          *
0228 006A 0283      CI   R3, 1          2-IF<LD FLAG. LE. 1> THEN
      006C 0001
0229 006E 1B0C      JH   LOA014
0230          *
0231          *
0232          *
0233          *
0234 0070 0A10      SLA  R0, 1          3-IF<CPL_PARM2 . NE. NULL>THN
0235 0072 1809      JOC  LOA012          4-BIAS= PARM2;
0236 0074 C0C3      MOV  R3, R3          3-END IF;
0237 0076 1605      JNE  LOA011          3-ELSE BIAS=DEFAULT_BIAS;
0238 0078 0201      LI   R1, BGNPXR          CHECK PRESENCE BIT
      007A 0000
0239 007C 0221      AI   R1, UFLBIA
      007E FE50
0240 0080 1003      JMP  LOA014
0241          LOA011 EQU  $
0242 0082 C520      MOV  @DFBIAS, *R4
      0084 0000
0243          LOA012 EQU  $
0244 0086 C074      MOV  *R4+, R1
0245          LOA014 EQU  $
0246          *
0247 0088 04E0      CLR  @LDCC          2-CALL SET LOADER WORKSPACE;
      008A 0000
0248 008C 0283      CI   R3, 2
      008E 0002
0249 0090 1307      JEQ  LOA030
0250          *
0251 0092 020B      LI   R11, LDIO          2-CALL SET_WORKSPACE<R11, LOC<L
      0094 0102
0252          *
0253 0096 C241      MOV  R1, R9          2-CALL SETWORKSPACE<R9, BIAS>
0254 0098 04C0      CLR  R0          SET FLAG FOR CASSETTE LOAD
0255          *
0256          *
0257 009A C0A0      MOV  @ROMLDR, R2          2-CALL LOADER<LOADER_IO,
      009C 0000          3-BIAS, CASS_FLAG>
0258 009E 0452      B    *R2
0259          LOA030 EQU  $
0260 00A0 06A0      BL   @ABSLDR
      00A2 0000
0261 00A4 0000      DATA LDTBL
0262          *****RETURN FROM ABSLOADER *****
0263 00A6 1023      JMP  LOA050          ERROR RETURN
0264          *
0265          *
0266          *
0267 00A8 00A0      MOV  @LDNMCC, R2
      00AA 0000
0268 00AC 1508      JEQ  LOA035
0269 00AE 06A0      BL   @PRCRLF
      00B0 0000
0270 00B2 1000      NOP
0271 00B4 020A      LI   R10, LDNMCC
      00B6 00AA
0272 00B8 06A0      BL   @PRNTC
      00BA 0000

```

```

0273 00BC 1000      NOP
0274      00BE' LOA035 EQU $
0275 00BE C0E0      MOV @LDTBL, R3
      00C0 00A4'

0276 ***** RETURN FROM CASSETTE LOADER *****
0277      00C2' RETLDR EQU $
0278 00C2 C360      MOV @SAVWP, R13
      00C4 002A'

0279 00C6 04DD      CLR *R13
0280 00C8 C0C3      MOV R3, R3
0281 00CA 1318      JEQ LOA060
0282      *
0283      *
0284 00CC C2AD      MOV @R14*2(R13), R10
      00CE 001C

0285 00D0 028A      CI R10, 3
      00D2 0003

0286 00D4 1313      JEQ LOA060
0287      *
0288 00D6 C803      MOV R3, @USRPC
      00D8 0000

0289      *
0290      *
0291 00DA C28A      MOV R10, R10
0292 00DC 160F      JNE LOA060
0293 00DE 05C3      INCT R3
0294 00E0 0693      BL *R3
0295 00E2 1005      JMP LOA050      ERROR RETURN
0296 00E4 C809      MOV R9, @USRPC
      00E6 00D8'

0297 00E8 C360      MOV @SAVWP, R13
      00EA 00C4'

0298 00EC 1007      JMP LOA060
0299      *
0300      00EE' LOA050 EQU $
0301      *
0302 00EE 020A      LI R10, LDERR
      00F0 2400

0303      00F2' LOA052 EQU $
0304 00F2 06A0      BL @ERROR
      00F4 0000

0305 00F6 C360      MOV @SAVWP, R13
      00F8 00EA'

0306 00FA 071D      SETO *R13
0307      *
0308      00FC' LUN060 EQU $
0309      *
0310      * *LINK TO PREV WKSP, RET CURR WKSP
0311 00FC 0420      BLWP @RR
      00FE 0000

0312 0100 045B      RT
    
```

3-IF PROGRAM_ENTRY . NE. 0 THEN
4-IF CALLER NE OVLY THEN

5-USER_PC = PROGRAM_ENTRY

5-IF CALLER EQ LOADUPFRONT THE
6-CALL UPFRONT LOADER

2-END; ELSE

3-CALL ERROR(<'LD00'>);

2-END;

2-RETURN;

0314	*			2-PROCEDURE LOADER_IO;
0315	*			3-/* LOADER_IO PROVIDES IO SUP
0316	*			3- FOR THE ROM LOADER. THE
0317	*			3- HAS BEEN PROCESSED BY THE
0318	*			3- DRIVER.
0319	*			3-*/
0320	*			
0321		0102'	LDIO EQU \$	
0322	*			MAY USE R8 THIS SUBROUTINE IS
0323	*			NON-STANDARD.
0324	*			5-CHAR_COUNT=CHAR_COUNT -1;
0325		0102 0620	DEC @LDCC	
		0104 008A'		
0326		0106 151F	JGT LDI050	
0327		0108 131E	JEQ LDI050	
0328	*			4-IF CHAR_COUNT =-1 THEN DO;
0329	*			5-LOAD_PRB_OPCODE = 'READ';
0330		010H 0208	LI R8, READ*LBYTE	
		010C 0900		
0331		010E D808	MOVB R8, @LDOPCD	
		0110 0060'		
0332	*			5-CALL SVC(SVC_CALL_BLOCK);
0333		0112 020H	LI R10, LDPRB	
		0114 0064'		
0334		0116 0420	BLWP @SYCALT	
		0118 0068'		
0335	*			5-IF ERROR THEN EXIT;
0336		011H 0220	MOVB @LDFLG, R8	
		011C 0000		
0337		011E 0A28	SLA R8, 2	
0338		0120 18E6	JOC LOA050	
0339	*			5-IF EOF THEN EXIT;
0340		0122 0A18	SLA R8, 1	
0341		0124 18CE	JOC RETLDR	
0342	*			5-CURR_BUFF_ADDR = LOADER_BUFF
0343		0126 C820	MOV @LDADDR, @LDCBA	
		0128 0000		
		012H 0000		
0344		012C C220	MOV @LDCC, R8	
		012E 0104'		
0345		0130 DA20	MOVB @CR, @LDBUF(R8)	
		0132 0000		
		0134 0000		
0346	*			5-OUTPUT LOAD ADDRESS TO LIGHT
0347		0136 C28C	MOV R12, R10	
0348		0138 020C	LI R12, CRUOFF	
		013H 0000		
0349		013C 3205	LDCR R5, 8	LOAD ADDR TO FP
0350		013E 06C5	SWPB R5	
0351		0140 3205	LDCR R5, 8	
0352		0142 06C5	SWPB R5	
0353		0144 C30A	MOV R10, R12	
0354	*			4-END;
0355		0146'	LDI050 EQU \$	
0356	*			4-CALL MOVE_BIT(RETURN_VALUE, C
0357		0146 C2A0	MOV @LDCBA, R10	
		0148 012A'		
0358	*			5-CURR_BUFF_ADDR = CURR_BUFF_A
0359		014H 05A0	INC @LDCBA	
		014C 0148'		

```

0360 014E 029A      MOVB *R10, R10
0361 0150 098A      SRL  R10, 8
0362 0152 01CA      A    R10, R7
0363 0154 028A      CI   R10, >3A
      0156 003A
0364 0158 130D      JEQ  LDI060
0365 015H 028A      CI   R10, >D
      015L 000D
0366 015E 1307      JEQ  GETRT
0367 0160 022A      AI   R10, ->30
      0162 FF00
0368 0164 028A      CI   R10, >11
      0166 0011
0369 0168 1102      JLT  GETRT
0370 016A 022A      AI   R10, -7
      016C FFF9
0371          016E' GETRT EQU  $
0372 016E 069B      BL   *R11
0373 0170 10C8      JMP  LDIO
0374 0172 10BD      JMP  LOA050
0375          *
0376          *
0377          *
0378          *
0379          0174' LDI060 EQU  $
0380 0174 06A0      BL   @LDIO
      0176 0102'
0381 0178 10FD      JMP  LDI060
0382          *
0383          *
0384          END

```

```

RETURN HERE FOR ERROR
3-END;
2-END LDIO;
2-DO FOREVER;
3-CALL GETCHAR;

```

```

FLUSH TO END OF RECORD

```

```

2-END
1-END LOADER_DRIVER;

```

```

NO ERRORS

```

* R	017A		0172	0176	0180	0184	0192	0194	0209	0241	0243
			0245	0259	0274	0277	0300	0303	0308	0321	0355
			0371	0379							
			0362								
HBSLUR	E	00H2	0114	0260							
HUL	E	0026	0131	0196							
AI			0239	0367	0370						
B			0258								
BGNPXR	E	007A	0133	0238							
BL			0215	0260	0269	0272	0294	0304	0372	0380	
BLWP			0196	0226	0311	0334					
C			0219								
CI			0228	0248	0285	0363	0365	0368			
CLDI	E	004A	0103	0215							
CLK			0173	0247	0254	0279					
CR	E	0152	0113	0345							
CRUOFF	E	015H	0126	0348							
CS	E	004E	0102	0216							
DATH			0261								
DEL			0325								
DEF			0094	0095	0096	0097					
DFBINS	E	0084	0124	0242							
DXUP			0154								
END			0384								
EQU			0137	0138	0139	0140	0141	0142	0143	0144	0145
			0146	0147	0148	0149	0150	0151	0152	0155	0156
			0157	0158	0159	0163	0172	0176	0180	0184	0192
			0194	0209	0241	0243	0245	0259	0274	0277	0300
			0303	0308	0321	0355	0371	0379			
EKKUR	E	00F4	0121	0304							
ETIBUF	E		0127								
GETKI	K	016E	0371	0366	0369						
IDI			0005								
INC			0359								
INCI			0293								
INII	E		0125								
JEU			0249	0268	0281	0286	0327	0364	0366		
JGI			0326								
JH			0229								
JLI			0369								
JMP			0174	0178	0190	0240	0263	0295	0298	0373	0374
			0381								
JNC			0206								
JNE			0220	0237	0292						
JUC			0235	0338	0341						
LBYTE		0100	0155	0201	0222	0330					
LDHBS	D	000H	0180	0095							
LDHDDR	E	0128	0109	0343							
LDBUF	E	0134	0112	0345							
LDCBH	E	014C	0111	0343	0357	0359					
LDCBBL	E		0122								
LDCU	E	012E	0110	0247	0325	0344					
LDCK			0349	0351							
LDERK		2400	0158	0302							
LDPLG	E	011C	0108	0336							
LDI050	R	0146	0355	0326	0327						
DI060	K	0174	0379	0364	0381						
DIU	K	0102	0321	0251	0373	0380					
LVLUN	E	0042	0107	0211							
LDNMUL	E	0086	0123	0267	0271						

LABEL	VALUE	DEFN	REFERENCES
KI			0312
KWP	E	0130	
SHYWP	E 00F8	0101	0197 0278 0297 0305
SETU			0306
SLH			0205 0208 0234 0337 0340
SKL			0214 0217 0361
SYL		0154	
SYCHLT	E 0118	0098	0226 0334
SYCSR	E 0018	0100	0187
SYLWP	E 0010	0099	0185
SWPB			0350 0352
TTL			0002
TRNHKM	E	0119	
UFLBIH	FE50	0163	0239
USKPL	E 00E6	0120	0288 0296

948935-9901 **


```

0064      *          CCNT = CCNT -1;
0065      *          END;
0066      *          CMDPTR.COMMAND_SVC_ROUTINE = COMMAND_ROU
0067      *          CPTR = CPTR + LEN(COMMAND_ENTRY);
0068      *          END;
0069      *          END; ELSE DO;
0070      *          /* INDICATE NO COMMANDS */
0071      *          LOC(TRANSIENT_AREA) = 0;
0072      *          END;
0073      *          REF'S AND DEF'S
0074      *
0075      *          REF  GETBUF
0076      *          REF  RETBUF
0077      *          REF  LWP
0078      *          REF  RWP
0079      *          REF  ACL
0080      *          REF  RR
0081      *          DEF  OVLY
0082      *          DEF  OVLRET
0083      *          REF  LOADOV
0084      *          REF  TRNARA
0085      *          REF  BGNPXR
0086      *          REF  BEGLAL
0087      *          REF  ENDLAL
0088      *          REF  CMDTBL
0089      *          REF  NOCMDS
0090      *          REF  PRCLRF
0091      *          REF  PRINT

```

START OF TRANSIENT AREA
BEGINNING OF PXR MONITOR
BEGIN OF LINK LOADER.
END OF LINK LOADER.

```

0092      *
0093      *WORKSPACE REGISTER DEFINITIONS
0094      *
0095      *          0000 R0      EQU  0
0096      *          0001 R1      EQU  1
0097      *          0002 R2      EQU  2
0098      *          0003 R3      EQU  3
0099      *          0004 R4      EQU  4
0100      *          0005 R5      EQU  5
0101      *          0006 R6      EQU  6
0102      *          0007 R7      EQU  7
0103      *          0008 R8      EQU  8
0104      *          0009 R9      EQU  9
0105      *          000A R10     EQU 10
0106      *          000B R11     EQU 11
0107      *          000C R12     EQU 12
0108      *          000D R13     EQU 13
0109      *          000E R14     EQU 14
0110      *          000F R15     EQU 15

```

```

0111      *
0112      *
0113      *
0114      *          1-PROCEDURE OVERLAY;
0115      *          0000' OVLY    EQU  $
0116      *          *ALLOC, COPY, LINK
0117      *          0000 0420    BLWP @ACL
0117      *          0002 0000

```

```

0118      *          2-/*
0119      *          2- OVERLAY IS RESPONSIBLE FOR
0120      *          2- MANAGING THE PX990 TRANSIE
0121      *          2- AREA. THIS INCLUDES;
0122      *          2- 1) DISABLING COMMANDS FOR

```

```

0123      *
0124      *
0125      *
0126      *
0127      *
0128      *
0129      *
0130      *
0131      *
0132      *
0133      *
0134      *
0135      *
0136      *
0137      *
0138      *
0139      0004  CMDRTN EQU 4
0140      0006  CMDLEN EQU 6
0141      0000  CDSTR  EQU 0
0142      0002  CDRTN  EQU 2
0143      *
0144      *      REGISTER ASSIGNMENTS
0145      *
0146      *      R10
0147      *      R9
0148      *      R8
0149      *      R7
0150      *      R6 - COMMAND ENTRY POINTER
0151      *      R5 - COMMAND TABLE POINTER
0152      *      R4 - TEMP
0153      *
0154      0004 0206      LI R6, TRNARA
0155      0006 0000
0156      *
0157      0008 C596      OVL10 EQU $
0158      000A 1310      MOV *R6, *R6
0159      *
0160      000C 0205      JEQ OVL50
0161      000E 0000
0162      *
0163      *
0164      0010 C120      LI R5, CMDTBL
0165      0012 0000
0166      *
0167      *
0168      0016 8556      MOV @NOCMDS, R4
0169      0018 1304
0170      *
0171      001A 0225      LI R5, CMDLEN
0172      001C 0006
0173      *
0174      *
0175      0020 10F9      MOV *R6, *R5
0176      0022 OVL40      JEQ OVL40
0177      0022 04E5      AI R5, CMDLEN
0178      0024 0004
0179      *
0180      *
0181      *
0182      *
0183      *
0184      *
0185      *
0186      *
0187      *
0188      *
0189      *
0190      *
0191      *
0192      *
0193      *
0194      *
0195      *
0196      *
0197      *
0198      *
0199      *
0200      *
0201      *
0202      *
0203      *
0204      *
0205      *
0206      *
0207      *
0208      *
0209      *
0210      *
0211      *
0212      *
0213      *
0214      *
0215      *
0216      *
0217      *
0218      *
0219      *
0220      *
0221      *
0222      *
0223      *
0224      *
0225      *
0226      *
0227      *
0228      *
0229      *
0230      *
0231      *
0232      *
0233      *
0234      *
0235      *
0236      *
0237      *
0238      *
0239      *
0240      *
0241      *
0242      *
0243      *
0244      *
0245      *
0246      *
0247      *
0248      *
0249      *
0250      *
0251      *
0252      *
0253      *
0254      *
0255      *
0256      *
0257      *
0258      *
0259      *
0260      *
0261      *
0262      *
0263      *
0264      *
0265      *
0266      *
0267      *
0268      *
0269      *
0270      *
0271      *
0272      *
0273      *
0274      *
0275      *
0276      *
0277      *
0278      *
0279      *
0280      *
0281      *
0282      *
0283      *
0284      *
0285      *
0286      *
0287      *
0288      *
0289      *
0290      *
0291      *
0292      *
0293      *
0294      *
0295      *
0296      *
0297      *
0298      *
0299      *
0300      *
0301      *
0302      *
0303      *
0304      *
0305      *
0306      *
0307      *
0308      *
0309      *
0310      *
0311      *
0312      *
0313      *
0314      *
0315      *
0316      *
0317      *
0318      *
0319      *
0320      *
0321      *
0322      *
0323      *
0324      *
0325      *
0326      *
0327      *
0328      *
0329      *
0330      *
0331      *
0332      *
0333      *
0334      *
0335      *
0336      *
0337      *
0338      *
0339      *
0340      *
0341      *
0342      *
0343      *
0344      *
0345      *
0346      *
0347      *
0348      *
0349      *
0350      *
0351      *
0352      *
0353      *
0354      *
0355      *
0356      *
0357      *
0358      *
0359      *
0360      *
0361      *
0362      *
0363      *
0364      *
0365      *
0366      *
0367      *
0368      *
0369      *
0370      *
0371      *
0372      *
0373      *
0374      *
0375      *
0376      *
0377      *
0378      *
0379      *
0380      *
0381      *
0382      *
0383      *
0384      *
0385      *
0386      *
0387      *
0388      *
0389      *
0390      *
0391      *
0392      *
0393      *
0394      *
0395      *
0396      *
0397      *
0398      *
0399      *
0400      *
0401      *
0402      *
0403      *
0404      *
0405      *
0406      *
0407      *
0408      *
0409      *
0410      *
0411      *
0412      *
0413      *
0414      *
0415      *
0416      *
0417      *
0418      *
0419      *
0420      *
0421      *
0422      *
0423      *
0424      *
0425      *
0426      *
0427      *
0428      *
0429      *
0430      *
0431      *
0432      *
0433      *
0434      *
0435      *
0436      *
0437      *
0438      *
0439      *
0440      *
0441      *
0442      *
0443      *
0444      *
0445      *
0446      *
0447      *
0448      *
0449      *
0450      *
0451      *
0452      *
0453      *
0454      *
0455      *
0456      *
0457      *
0458      *
0459      *
0460      *
0461      *
0462      *
0463      *
0464      *
0465      *
0466      *
0467      *
0468      *
0469      *
0470      *
0471      *
0472      *
0473      *
0474      *
0475      *
0476      *
0477      *
0478      *
0479      *
0480      *
0481      *
0482      *
0483      *
0484      *
0485      *
0486      *
0487      *
0488      *
0489      *
0490      *
0491      *
0492      *
0493      *
0494      *
0495      *
0496      *
0497      *
0498      *
0499      *
0500      *
0501      *
0502      *
0503      *
0504      *
0505      *
0506      *
0507      *
0508      *
0509      *
0510      *
0511      *
0512      *
0513      *
0514      *
0515      *
0516      *
0517      *
0518      *
0519      *
0520      *
0521      *
0522      *
0523      *
0524      *
0525      *
0526      *
0527      *
0528      *
0529      *
0530      *
0531      *
0532      *
0533      *
0534      *
0535      *
0536      *
0537      *
0538      *
0539      *
0540      *
0541      *
0542      *
0543      *
0544      *
0545      *
0546      *
0547      *
0548      *
0549      *
0550      *
0551      *
0552      *
0553      *
0554      *
0555      *
0556      *
0557      *
0558      *
0559      *
0560      *
0561      *
0562      *
0563      *
0564      *
0565      *
0566      *
0567      *
0568      *
0569      *
0570      *
0571      *
0572      *
0573      *
0574      *
0575      *
0576      *
0577      *
0578      *
0579      *
0580      *
0581      *
0582      *
0583      *
0584      *
0585      *
0586      *
0587      *
0588      *
0589      *
0590      *
0591      *
0592      *
0593      *
0594      *
0595      *
0596      *
0597      *
0598      *
0599      *
0600      *
0601      *
0602      *
0603      *
0604      *
0605      *
0606      *
0607      *
0608      *
0609      *
0610      *
0611      *
0612      *
0613      *
0614      *
0615      *
0616      *
0617      *
0618      *
0619      *
0620      *
0621      *
0622      *
0623      *
0624      *
0625      *
0626      *
0627      *
0628      *
0629      *
0630      *
0631      *
0632      *
0633      *
0634      *
0635      *
0636      *
0637      *
0638      *
0639      *
0640      *
0641      *
0642      *
0643      *
0644      *
0645      *
0646      *
0647      *
0648      *
0649      *
0650      *
0651      *
0652      *
0653      *
0654      *
0655      *
0656      *
0657      *
0658      *
0659      *
0660      *
0661      *
0662      *
0663      *
0664      *
0665      *
0666      *
0667      *
0668      *
0669      *
0670      *
0671      *
0672      *
0673      *
0674      *
0675      *
0676      *
0677      *
0678      *
0679      *
0680      *
0681      *
0682      *
0683      *
0684      *
0685      *
0686      *
0687      *
0688      *
0689      *
0690      *
0691      *
0692      *
0693      *
0694      *
0695      *
0696      *
0697      *
0698      *
0699      *
0700      *
0701      *
0702      *
0703      *
0704      *
0705      *
0706      *
0707      *
0708      *
0709      *
0710      *
0711      *
0712      *
0713      *
0714      *
0715      *
0716      *
0717      *
0718      *
0719      *
0720      *
0721      *
0722      *
0723      *
0724      *
0725      *
0726      *
0727      *
0728      *
0729      *
0730      *
0731      *
0732      *
0733      *
0734      *
0735      *
0736      *
0737      *
0738      *
0739      *
0740      *
0741      *
0742      *
0743      *
0744      *
0745      *
0746      *
0747      *
0748      *
0749      *
0750      *
0751      *
0752      *
0753      *
0754      *
0755      *
0756      *
0757      *
0758      *
0759      *
0760      *
0761      *
0762      *
0763      *
0764      *
0765      *
0766      *
0767      *
0768      *
0769      *
0770      *
0771      *
0772      *
0773      *
0774      *
0775      *
0776      *
0777      *
0778      *
0779      *
0780      *
0781      *
0782      *
0783      *
0784      *
0785      *
0786      *
0787      *
0788      *
0789      *
0790      *
0791      *
0792      *
0793      *
0794      *
0795      *
0796      *
0797      *
0798      *
0799      *
0800      *
0801      *
0802      *
0803      *
0804      *
0805      *
0806      *
0807      *
0808      *
0809      *
0810      *
0811      *
0812      *
0813      *
0814      *
0815      *
0816      *
0817      *
0818      *
0819      *
0820      *
0821      *
0822      *
0823      *
0824      *
0825      *
0826      *
0827      *
0828      *
0829      *
0830      *
0831      *
0832      *
0833      *
0834      *
0835      *
0836      *
0837      *
0838      *
0839      *
0840      *
0841      *
0842      *
0843      *
0844      *
0845      *
0846      *
0847      *
0848      *
0849      *
0850      *
0851      *
0852      *
0853      *
0854      *
0855      *
0856      *
0857      *
0858      *
0859      *
0860      *
0861      *
0862      *
0863      *
0864      *
0865      *
0866      *
0867      *
0868      *
0869      *
0870      *
0871      *
0872      *
0873      *
0874      *
0875      *
0876      *
0877      *
0878      *
0879      *
0880      *
0881      *
0882      *
0883      *
0884      *
0885      *
0886      *
0887      *
0888      *
0889      *
0890      *
0891      *
0892      *
0893      *
0894      *
0895      *
0896      *
0897      *
0898      *
0899      *
0900      *
0901      *
0902      *
0903      *
0904      *
0905      *
0906      *
0907      *
0908      *
0909      *
0910      *
0911      *
0912      *
0913      *
0914      *
0915      *
0916      *
0917      *
0918      *
0919      *
0920      *
0921      *
0922      *
0923      *
0924      *
0925      *
0926      *
0927      *
0928      *
0929      *
0930      *
0931      *
0932      *
0933      *
0934      *
0935      *
0936      *
0937      *
0938      *
0939      *
0940      *
0941      *
0942      *
0943      *
0944      *
0945      *
0946      *
0947      *
0948      *
0949      *
0950      *
0951      *
0952      *
0953      *
0954      *
0955      *
0956      *
0957      *
0958      *
0959      *
0960      *
0961      *
0962      *
0963      *
0964      *
0965      *
0966      *
0967      *
0968      *
0969      *
0970      *
0971      *
0972      *
0973      *
0974      *
0975      *
0976      *
0977      *
0978      *
0979      *
0980      *
0981      *
0982      *
0983      *
0984      *
0985      *
0986      *
0987      *
0988      *
0989      *
0990      *
0991      *
0992      *
0993      *
0994      *
0995      *
0996      *
0997      *
0998      *
0999      *
1000      *

```

```

2- CURRENT AREA.
2- 2) CALLING THE LOADER TO
THE NEW OVERLAY.
2- 3) ENABLING THE COMMANDS
NEW OVERLAY.
2- 4) BUILDING A TRANSFER VE
USE BY THE OVERLAY IN
THE SYSTEM.
2-*/
2-DECLARE
3-TRANSIENT_AREA(700) FIXED(16
3-VECTOR(2) PTR CONTROL TPTR;
2-DECLARE
3-1 COMMAND_ENTRY CONTROL CPT
4-2 COMMAND_STRING CHAR(2);
4-2 COMMAND_ROUT POINTER;

COMMAND TEXT STRING
COMMAND SERVICE ROUTINE.

```

```

REGISTER ASSIGNMENTS

R10
R9
R8
R7
R6 - COMMAND ENTRY POINTER
R5 - COMMAND TABLE POINTER
R4 - TEMP

2-CPTR = LOC(TRANSIENT_AREA) +

2-DO WHILE COMMAND_ENTRY . NE.

3-CMDPTR = LOC(COMMAND_TABLE);

3-CCNT = NO_CMDS;

3-DO WHILE CCNT>0;

4-IF COMMAND_STRING . NE. CMDPTR
4-THEN DO;

4-CMDPTR = CMDPTR + LEN(COMMAN

4-CCNT = CCNT -1;

3-END;

```

```

0178 * 3-CPTR = CPTR + LEN(COMMAND_LEN
0179 0026 0226 AI R6, CDRTN+2
      0028 0004
0180 * 2-END
0181 002A 10EE JMP OVLY10
0182 002C' OVLY50 EQU $
0183 ** LOAD OVERLAY
0184 * 2-BIAS= LOC(BGNPXR-LENGTH OF L
0185 002C 0201 LI R1, BGNPXR
      002E 0000
0186 * 2-CALL LOAD(FLAG, BIAS, ERROR)
0187 0030 06A0 BL @LOADOV
      0032 0000
0188 0034' OVLRET EQU $
0189 * 2-IF LOAD SUCCESSFUL THEN DO;
0190 0034 C000 MOV R0, R0
0191 0036 161E JNE OVLZ50
0192 *
0193 * 3-CPTR = LOC(TRANSIENT_AREA)
0194 0038 0206 LI R6, TRNARA
      003A 0006'
0195 * 3-DO WHILE COMMAND ENTRY . NE.
0196 003C' OVLZ10 EQU $
0197 003C C596 MOV *R6, *R6
0198 003E 131C JEQ OVLZ60
0199 * 4-CMDPTR = LOC(COMMAND_TABLE);
0200 0040 0205 LI R5, CMDTBL
      0042 000E'
0201 * 4-CCNT = NO_CMDS
0202 0044 C120 MOV @NOCMDS, R4
      0046 0012'
0203 * 4-DO WHILE CCNT>0;
0204 0048' OVLZ20 EQU $
0205 0048 1315 JEQ OVLZ50
0206 * 5-IF COMMAND_STRING . EQ. CMDPT
0207 * 5-THEN DO;
0208 004A 8556 C *R6, *R5
0209 004C 1304 JEQ OVLZ30
0210 * 5-CMDPTR = CMDPTR + LEN(COMMAN
0211 004E 0225 AI R5, CMDLEN
      0050 0006
0212 * 5-CCNT = CCNT -1;
0213 0052 0604 DEC R4
0214 * 4-END;
0215 0054 10F9 JMP OVLZ20
0216 0056' OVLZ30 EQU $
0217 * 6-CMDPTR. COMMAND_SVC_ROUTINE =
0218 0056 C966 MOV @CDRTN(R6), @CMDRTN(R5)
      0058 0002
      005A 0004
0219 005C 06A0 BL @PRCRLF
      005E 0000
0220 0060 1000 NOP
0221 0062 C285 MOV R5, R10
0222 0064 0209 LI R9, 2
      0066 0002
0223 0068 06A0 BL @PRINT
      006A 0000
0224 006C 1000 NOP
0225 * 4-CPTR = CPTR + LEN(COMMAND_LEN

```



```
0226 006E 0226          AI  R6.CDRTN+2
      0070 0004
0227          *
0228 0072 10E4          JMP  OVLZ10          3-END;
0229          *
0230          *
0231 0074' OVLZ50 EQU $          2-END; ELSE DO;
0232          *
0233 0074 04E0          CLR  @TRNARA          3-/* INDICATE NO COMMANDS */
      0076 003A'          *
0234          *
0235          *
0236 0078' OVLZ60 EQU $          2-END;
      007A 0000          BLWP @RR
0237 007C 045B          RT
0238          END
NO ERRORS
```

SDSMAC 947075 *D 15:31:20 TUESDAY, FEB 22, 1977. PAGE 0007

OVERLAY LABEL	VALUE	DEFN	REFERENCES	948935-9901	**							
\$	R	007E		0115	0156	0164	0176	0182	0188	0196	0204	0216
				0230	0235							
ACL	E	0002	0079	0117								
AI				0171	0179	0211	0226					
BEG LAL	E		0086									
BGNPXR	E	002E	0085	0185								
BL				0187	0219	0223						
BLWP				0117	0236							
C				0168	0208							
CDRTN		0002	0142	0179	0218	0226						
CDSTR		0000	0141									
CLR				0177	0233							
CMDLEN		0006	0140	0171	0211							
CMDRTN		0004	0139	0177	0218							
CMDTBL	E	0042	0088	0160	0200							
DEC				0173	0213							
DEF				0081	0082							
END				0238								
ENDLAL	E		0087									
EQU				0095	0096	0097	0098	0099	0100	0101	0102	0103
				0104	0105	0106	0107	0108	0109	0110	0115	0139
				0140	0141	0142	0156	0164	0176	0182	0188	0196
				0204	0216	0230	0235					
GETBUF	E		0075									
IDT				0004								
JEQ				0158	0165	0169	0198	0205	0209			
JMP				0175	0181	0215	0228					
JNE				0191								
LI				0154	0160	0185	0194	0200	0222			
LOADOV	E	0032	0083	0187								
LWP	E		0077									
MOV				0157	0162	0190	0197	0202	0218	0221		
NOCMDS	E	0046	0089	0162	0202							
NOP				0220	0224							
OVLRET	D	0034	0188	0082								
OVLV	D	0000	0115	0081								
OVLV10	R	0008	0156	0181								
OVLV20	R	0014	0164	0175								
OVLV40	R	0022	0176	0169								
OVLV50	R	002C	0182	0158	0165							
OVLZ10	R	003C	0196	0228								
OVLZ20	R	0048	0204	0215								
OVLZ30	R	0056	0216	0209								
OVLZ50	R	0074	0230	0191	0205							
OVLZ60	R	0078	0235	0198								
PAGE				0001	0003							
PRCRLF	E	005E	0090	0219								
PRINT	E	006A	0091	0223								
R0		0000	0095	0190	0190							
R1		0001	0096	0185								
R10		000A	0105	0221								
R11		000B	0106									
R12		000C	0107									
R13		000D	0108									
R14		000E	0109									
R15		000F	0110									
R2		0002	0097									
R3		0003	0098									
R4		0004	0099	0162	0173	0202	0213					
R5		0005	0100	0160	0168	0171	0177	0200	0208	0211	0218	0221

OVERLAY
LABEL

SDSMAC 947075 *D 15:31:20 TUESDAY, FEB 22, 1977.

VALUE DEFN REFERENCES 948935-9901 **

PAGE 0008

R6	0006	0101	0154	0157	0157	0168	0179	0194	0197	0197	0208
			0218	0226							
R7	0007	0102									
R8	0008	0103									
R9	0009	0104	0222								
REF			0075	0076	0077	0078	0079	0080	0083	0084	0085
			0086	0087	0088	0089	0090	0091			
RETBUF	E	0076									
RR	E	007A	0080	0236							
RT			0237								
RWP	E	0078									
TITL			0002								
TRNARA	E	0076	0084	0154	0194	0233					


```

0003          IDT  <PROMLD>
0004      * TITLE:  PROMLD
0005      *          PROM LOAD
0006      * REVISION:
0007      *          MODIFIED TO RUN UNDER PXRMAE SYSTEM
0008      * COMPUTER: 990,ASM
0009      * ABSTRACT: THIS PROGRAM WILL LOAD THE PROM PROGRAMMER INTO
0010      *          MEMORY. ONE PART IS LOADED IN THE OVERLAY AREA
0011      *          BY A CALL TO THE OVERLAY DRIVER. THE SECOND PART
0012      *          IS LOADED INTO THE USER AREA BY CALLING THE LOAD
0013      *          DRIVER AND SIMULATING AN OVERLAY CALL.
0014      * CALLING SEQUENCE:
0015      *          CALLED FROM COMMAND STRING PROCESSOR
0016      *          ENTRY - R10 POINTS TO COMMAND PARM LIST
0017      *          DESTROYS R1, R8, R9, R15, R14
0018      *
0019          DEF  PL
0020          REF  ENDBUF, OVLY, LOADOV, PGBIAS
0021          REF  PROGSZ
0022      *
0023      0001 R1    EQU  1
0024      0008 R8    EQU  8
0025      0009 R9    EQU  9
0026      000A R10  EQU 10
0027      000B R11  EQU 11
0028      000F R15  EQU 15
0029      *
0030      *          -1-PROCEDURE FROM_LOAD(CPL_POINTER)
0031      0000 PL    EQU  $
0032      0000 C3CB MOV  R11, R15          SAVE RETURN ADDRESS
0033      *
0034      0002 06A0 BL   @OVLY
0035          0004 0000
0036      *          -2-DFBIAS = LOC(INIT) - PROGLSIZE
0037      0006 0201 LI   R1, ENDBUF
0038          0008 0000
0039      000A C220 MOV  @PROGSZ, R8
0040          000C 0000
0041      000E 6048 S    R8, R1
0042      *          -2-POINTER = CPL_POINTER
0043      0010 C20A MOV  R10, R8
0044      0012 C278 MOV  *R8+, R9
0045      0014 0A99 SLA  R9, 9
0046      *          -2-IF PARM(1) .NE. NULL THEN
0047      0016 1701 JNC  PL1
0048      *          -3-POINTER = POINTER + 2
0049      0018 05C8 INCT R8
0050      001A 001A PL1  EQU  $
0051          0A19 SLA  R9, 1
0052      *          -2-IF PARM(2) .NE. NULL THEN
0053      001C 1701 JNC  PL2
0054      *          -3- BIAS = LOC(POINTER)
0055      001E C058 MOV  *R8, R1
0056          0020 PL2  EQU  $
0057      0020 C801 MOV  R1, @PGBIAS          STORE PROG BIAS IN OVERLAY
0058          0022 0000
0059      *          -2-CALL LOADOV(CPL_POINTER, BIAS)
0060      0024 06A0 BL   @LOADOV
0061          0026 0000
0062      *          -1-END FROM_LOAD

```

PROMLD SDSMAC 947075 *B 14:30:09 TUESDAY, JAN 04, 1977.
LOAD FROM PROGRAMMER 948937-9901**

PAGE 000

0058 0028 045F
0059.
NO ERRORS

B *R15
END

LABEL	VALUE	DEFN	REFERENCES
\$	R 002A		0031 0047 0053
B			0058
BL			0034 0056
DEF			0019
END			0059
ENDBUF	E 0020		0036
EGU			0023 0024 0025 0026 0027 0028 0031 0047 0053
IDT			0003
INCT			0046
JNC			0044 0050
L1			0036
LOADOV	E 0020		0056
MOV			0032 0037 0040 0041 0052 0054
OVLY	E 0020		0034
PAGE			0002
PGBIAS	E 0020		0054
PL	D 0000	0031	0019
PL1	R 001A	0047	0044
PL2	R 0020	0053	0050
PROGSZ	E 0021		0037
R1	0001	0023	0036 0038 0052 0054
R10	000A	0026	0040
R11	000B	0027	0032
R15	000F	0028	0032 0058
R8	0008	0024	0037 0038 0040 0041 0046 0052
R9	0009	0025	0041 0042 0048
REF			0020 0021
S			0038
SLA			0042 0048
TITL			0001

MINIT SDSMAC 947075 *D
** NEW INITIATE MON**

11:53:54 FRIDAY, FEB 18, 1977.
948938-9901**

PAGE 0001A

** NEW INITIATE MON**

948938-9901**

```

0004          IDT  'MINIT'
0005      * TITLE:   DEFINED
0006      * COMPUTER: 990,ASM
0007      * ORIGINAL
0008      *REVISION: 9/5/76 TO SUPPORT PAPER TAPE
0009      * ABSTRACT: INITIALIZES THE MONITOR WHENEVER IT IS
0010      *              LOADED OR RESTARTED. BRANCHES TO
0011      *              COMMAND STRING PROCESSOR.
0012      * CALLING SEQUENCE:
0013      *              LOAD OR RESTART MONITOR
0014      *              OR
0015      *              B   @INIT
0016      *
0017          DEF  INIT
0018          REF  CRUPRT
0019          REF  CSP
0020          REF  SVCSR
0021          REF  SVCWP
0022          REF  DC7331
0023          REF  DCTTYT
0024          REF  EI
0025          REF  FREMEM
0026          REF  GETBUF
0027          REF  INIMEM
0028          REF  FPSTRT
0029          1FC0 CLRERR EQU  >1FC0
0030      *
0031          0000 WRTPRT EQU  0
0032      *
0033          0001 R1     EQU  1
0034          000C R12    EQU  12
0035          0000 INIT   EQU  $
0036          0000 02E0 LWPI FREMEM
0037          0002 0000
0038          0004 0300
0039          0006 0002
0038      *
0039      * CLEAR PARITY ERRORS
0040      *
0041          0008 0360 RSET
0042      *
0043      * FRONT PANEL INTERRUPT VECTOR
0044      *
0045          000A 0201 LI   R1,EI
0046          000C 0000
0047          000E C801 MOV  R1,@>A
0048          0010 000A
0047      *
0048      * TURN OFF PROTECT VIOLATION FLAG AND CLEAR PROTECT RE
0049      *
0050          0012 020C LI   R12,CRUPRT
0051          0014 0000
0052          0016 1D00 SBO  WRTPRT
0053      *
0054      * INITIALIZE 33
0055          0018 C320 MOV  @DCTTYT,R12 MOVE ASR 33 CRU BASE
0056          001A 0000
0057          001C 1D09 SBO  >9 DATA TERMINAL READY
0058          001E 1D0B SBO  >B CLEAR READ REQUEST

```

```
0058 0020 1D0C          SBO >C          CLEAR WRITE REQUEST
0059 0022 1E0F          SBZ >F          DIAGNOSTIC MODE
0060                    *
0061                    *      INITIALIZE 733
0062                    *
0063      0024' INI733 EQU $
0064 0024 C320          MOV @DC7331, R12  MOVE CRU BASE
      0026 0000
0065 0028 1D09          SBO >9          DATA TERMINAL READY
0066 002A 1D0A          SBO >A          CLEAR REQUEST TO SEND
0067 002C 1D0B          SBO >B          CLEAR READ REQUEST
0068 002E 1D0C          SBO >C          CLEAR WRITE REQUEST
0069                    *
0070                    *      INITIALIZE MEMORY CHAINS
0071                    *
0072 0030 0420          BLWP @INIMEM
      0032 0000
0073 0034 0420          BLWP @GETBUF
      0036 0000
0074 0038 0460          B @CSP
      003A 0000
0075 0000'            END INIT
NO ERRORS
```


MPXIOS SDSMAC 947075 *B
** NEW I/O DISPATCH MODULE **

12:14:10 TUESDAY, SEP 07, 1976.
948939-9901*

PAGE 0001A

0004 IDT 'MPXI05'
0005 *
0006 * TITLE: IODIS
0007 *
0008 * ORIGINAL:
0009 *REVISIONS: 09/03/76 TO RECOGNIZE TTY I/O CALLS
0010 *
0011 * COMPUTER: 990
0012 *
0013 * ABSTRACT: THIS ROUTINE DOES I/O CALLS FOR PX990 WITH
0014 * DEVICES: ASR, CASSETTE, ASR 33 PAPER TAPE PUNCH/
0015 * READER A GROUPING OF SUBROUTINES WHICH
0016 * CONSTITUTE THE PX99-1 IO EXECUTIVE
0017 *
0018 * TABLE OF CONTENTS:
0019 * 1. LUNO/DEVICE TYPE/DEVICE NAME CONVERSION
0020 * A. DEVICE NAME TO DEVICE TYPE
0021 * B. SET DEVICE TYPE IN LUNO TABLE
0022 * C. LUNO TO DEVICE TYPE
0023 * 2. IO SERVICE DISPATCHER
0024 *

```

0026      *
0027      *      WORKSPACE REGISTER DEFINITIONS
0028      *
0029      0000 R0      EQU  0
0030      0001 R1      EQU  1
0031      0002 R2      EQU  2
0032      0003 R3      EQU  3
0033      0004 R4      EQU  4
0034      0005 R5      EQU  5
0035      0006 R6      EQU  6
0036      0007 R7      EQU  7
0037      0008 R8      EQU  8
0038      0009 R9      EQU  9
0039      000A R10     EQU 10
0040      000B R11     EQU 11
0041      000C R12     EQU 12
0042      000D R13     EQU 13
0043      000E R14     EQU 14
0044      000F R15     EQU 15

```

```

0045      *
0046      *      REFERENCES
0047      *

```

```

0048      REF  ACL
0049      REF  DCTTYT
0050      REF  DC7331
0051      REF  LGRASC
0052      REF  LGWASC
0053      REF  CSOPEN
0054      REF  CSBACK
0055      REF  CSFWD
0056      REF  CSRASC
0057      REF  CSRWND
0058      REF  CSUNLD
0059      REF  CSWASC
0060      REF  CSWEOF
0061      REF  PTOPEM
0062      REF  PTCLOS
0063      REF  PTCEOF
0064      REF  PTOPRW
0065      REF  PTDIRT
0066      REF  PTWASC
0067      REF  PTWEOF
0068      REF  PROPEN
0069      REF  PRCLOS
0070      REF  PRDIRT
0071      REF  PRDASC
0072      REF  PTRILL
0073      REF  RR

```

```

0074      *
0075      *      DEFINITIONS
0076      *

```

```

0077      DEF  CDNDT
0078      DEF  CLDT
0079      DEF  CMDMSK      CHARACTER MODE
0080      DEF  EFMASK      END-OF-FILE FLAG
0081      DEF  IO          IO SERVICE DISPATCHER
0082      DEF  SETLUN
0083      DEF  UEMASK      UNRECOV. ERROR
0084      DEF  UEFMSK      UNRECOV ERROR AND EOF FLG
0085      DEF  LOG, DUM, CS, PTP, PTR

```


0086		*				
0087	0000	NDV1YP	EQU	5		NUMBER OF DEVICES
0088		*				
0089	0010	NUMLUN	EQU	16		MAX. NO. LUNO
0090		*				
0091	0000	DYNHME	EQU	#		DEVICE NAME TABLE
0092	0000	4C	TEXT	'LOG'		
	0001	4F				
	0002	47				
0093	0003	44	TEXT	'DUM'		
	0004	55				
	0005	4D				
0094	0006	43	TEXT	'CS1'		
	0007	53				
	0008	31				
0095	0009	43	TEXT	'CS2'		
	000A	53				
	000B	32				
0096	000C	50	TEXT	'PTP'		
	000D	54				
	000E	50				
0097	000F	50	TEXT	'PTR'		
	0010	54				
	0011	52				
0098	007F	DUM	EQU	127		
0099	0000	LOG	EQU	0		
0100	0001	CS	EQU	1		
0101	0002	PTP	EQU	2		
0102	0003	PTR	EQU	3		
0103	0006	NUMDEV	EQU	6		TOTAL NUMBER OF PERIPHERALS CONNECTED
0104		*				
0105		*				
0106	0012	DVNDVT	EQU	#		DEVICE NAME TO DEVICE TYPE
0107	0012	00	BYTE	LOG		
0108	0013	7F	BYTE	DUM		
0109	0014	01	BYTE	CS		
0110	0015	01	BYTE	CS		
0111	0016	02	BYTE	PTP		
0112	0017	03	BYTE	PTR		
0113		*				
0114	0018	DVNUNT	EQU	#		DEVICE NAME TO UNIT NUMBER
0115	0018	00	BYTE	0		
0116	0019	00	BYTE	0		
0117	001A	00	BYTE	0		
0118	001B	01	BYTE	1		
0119	001C	00	BYTE	0		
0120	001D	00	BYTE	0		
0121		*				
0122	001E	LUNDVT	EQU	#		LOGICAL UNIT NO TO DEV TYPE
0123	001E	00	BYTE	LOG, DUM, DUM, DUM		
	001F	7F				
	0020	7F				
	0021	7F				
0124	0022	7F	BYTE	DUM, DUM, LOG, CS		
	0023	7F				
	0024	00				
	0025	01				
0125	0026	01	BYTE	CS, DUM, DUM, DUM		
	0027	7F				
	0028	7F				

0029 7F
 0126 002A 7F BYTE DUM, DUM, DUM, DUM
 002B 7F
 002C 7F
 002D 7F

0127 *
 0128 002E' LUNUNT EQU \$ LOGICAL UNIT NO TO PHYS UNIT
 0129 002E 00 BYTE 0, 0, 0, 0
 002F 00
 0030 00
 0031 00
 0130 0032 00 BYTE 0, 0, 0, 0
 0033 00
 0034 00
 0035 00
 0131 0036 01 BYTE 1, 0, 0, 0
 0037 00
 0038 00
 0039 00
 0132 003A 00 BYTE 0, 0, 0, 0
 003B 00
 003C 00
 003D 00

0133 *
 0134 003E EVEN
 0135 0100 LBYTE EQU >100 MOVE BYTE TO LEFT BYTE

```

0137      *
0138      * TITLE:      CDNDT
0139      *           CONVERT DEVICE NAME TO DEVICE TY&E/UNIT NO.
0140      * REVISION
0141      *           ORIGINAL
0142      * COMPUTER:  990
0143      * ABSTRACT: THIS ROUTINE CONVERTS A THREE-CHARACTER
0144      *           DEVICE NAME TO A TYPE AND PHYSICAL
0145      *           UNIT NUMBER.
0146      * CALLING SEQUENCE:
0147      *           MOV <ADDRESS OF DEVICE NAME STRING>,R10
0148      *           BL   @CDNDT
0149      *           UPON RETURN:
0150      *           R9L: DEVICE TYPE (ERROR IF = -1)
0151      *           R8L: PHYSICAL UNIT NUMBER
0152      *
0153      003E' CDNDT EQU $           CONVERT DVNAME TO DVTYPE
0154      003E 0209          LI R9, NUMDEV      SET LOOP VARIABLE
0155      0040 0006
0155      0042' CDNDT2 EQU $
0156      0042 0609          DEC R9           END OF LOOP?
0157      0044 1110          JLT CDNDT8       YES
0158      0046 C009          MOV R9, R0       DEVELOP DVNAME
0159      0048 A009          A R9, R0
0160      004A A009          A R9, R0
0161      004C 0220          AI R0, DVNAME
0162      004E 0000'
0162      *
0163      0050 C04A          MOV R10, R1          IF NOT EQUAL
0164      0052 9C70          CB *R0+, *R1+
0165      0054 16F6          JNE CDNDT2
0166      0056 9C70          CB *R0+, *R1+
0167      0058 16F4          JNE CDNDT2
0168      005A 9450          CB *R0, *R1
0169      005C 16F2          JNE CDNDT2
0170      *
0171      005E D229          MOVEB @DVNUNT(R9), R8      SET UNIT NUMBER
0172      0060 0018'
0172      0062 D269          MOVEB @DVNDVT(R9), R9
0173      0064 0012'
0173      *
0174      0066' CDNDT8 EQU $
0175      0066 045B          RT           RETURN

```

```

0177      *
0178      * TITLE:      SETLUN
0179      *           SET DEVICE TYPE/UNIT IN LUNO TABLE
0180      * REVISION:
0181      *           ORIGINAL
0182      * COMPUTER:  990
0183      * ABSTRACT:  THIS ROUTINE SETS DEVICE TYPE AND
0184      *           PHYSICAL UNIT NUMBER FOR THE INDICATED LOGICAL
0185      *           UNIT NUMBER.
0186      * CALLING SEQUENCE:
0187      *           MOV  <LUNO>,R10
0188      *           MOVB <DEVICE TYPE>,R9
0189      *           MOVB <UNIT NUMBER>,R8
0190      *           BL   @SETLUN
0191      *           UPON RETURN:
0192      *           R7: ERROR CODE (0 = NO ERROR, -1 = ERROR)
0193      *
0194      0068' SETLUN EQU  $
0195      0068 0707      SETO  R7          ERROR CODE
0196      006A C28A      MOV   R10,R10
0197      006C 1308      JEQ   SLUN90
0198      006E 028A      CI    R10,NUMLUN
0199      0070 0010
0199      0072 1405      JHE   SLUN90
0200      0074 DA89      MOVB  R9,@LUNDVT(R10)  SET DEVICE TYPE
0200      0076 001E'
0201      0078 DA88      MOVB  R8,@LUNUNT(R10)  SET PHYSICAL UNIT NUMBER
0201      007A 002E'
0202      007C 04C7      CLR   R7          CLEAR ERROR CODE
0203      007E' SLUN90 EQU  $
0204      007E 045B      RT          RETURN

```

```

0206      *
0207      * TITLE:      CLDT
0208      *           CONVERT LUN0 TO DEVICE TYPE AND UNIT
0209      * REVISION:
0210      *           ORIGINAL
0211      * COMPUTER:  990
0212      * ABSTRACT: THIS ROUTINE RETURNS THE DEVICE TYPE AND
0213      *           PHYSICAL UNIT NUMBER FOR THE GIVEN LUN0.
0214      * CALLING SEQUENCE:
0215      *           MOV <LUN0>,R10
0216      *           BL @CLDT
0217      *           UPON RETURN:
0218      *           R9L: DEC
0219      *           R9L: DEVICE TYPE (ERROR ==-1)
0220      *           R8L: UNOT NUMBER
0221      *
0222      0080' CLDT EQU $           CONVERTS LUN TO DEVICE TYPE
0223      0080 0209      LI R9,DUM*LBYTE
0224      0082 7F00
0224      0084 028A      CI R10,NUMLUN      IN RANGE?
0224      0086 0010
0225      0088 1404      JHE CLDT90      YES
0226      008A D26A      MOVE @LUNDVT(R10),R9
0226      008C 001E'
0227      008E D22A      MOVE @LUNUNT(R10),R8
0227      0090 002E'
0228      0092' CLDT90 EQU $
0229      0092 045B      RT           RETURN

```

```

0231      *
0232      *      PRB DISPLACEMENT EQUATES:
0233      *
0234      0000 IOOP EQU 0      IO OP CODE
0235      0001 LUND EQU 1      LOGICAL UNIT NUMBER
0236      0002 SYSFLG EQU 2    SYSTEM FLAGS
0237      0003 USEFLG EQU 3    USER FLAGS
0238      0004 BFADDR EQU 4    BUFFER ADDRESS
0239      0006 BFLNTH EQU 6    BUFFER LENGTH
0240      0008 CCOUNT EQU 8   CHARACTER COUNT
0241      *      FLAG SHIFT COUNTS:
0242      000F UE EQU 16-1     UNRECOV. ERROR
0243      000E EOFLG EQU 16-2  EOD-/F-FILE FLG
0244      000D CHRMOD EQU 16-3  CHARACTER MODE
0245      *      FLAG SET MASKS:
0246 0094 40 UEMASK BYTE >40
0247 0095 20 EFMASK BYTE >20
0248 0096 10 CMDMSK BYTE >10
0249 0097 60 UEFMSK BYTE >40+>20
0250      *
0251      0000 OCTCRU EQU 0     OCT MAPPING TABLES
0252      *
0253      *      MAP DEVICE TYPE      (ONE ENTRY PER DEV TYPE)
0254      *
0255      0098' OCTDT EQU $
0256 0098 04      BYTE UTLOG-OCTDT      LOG
0257 0099 06      BYTE UTCST-OCTDT      CASSETTE
0258 009A 0A      BYTE UTPTP-OCTDT      TAPE PUNCH
0259 009B 0C      BYTE UTPTR-OCTDT      TAPE READER
0260 009C      EVEN
0261      *
0262      *      MAP UNIT TO OCT
0263      *
0264      009C' UTLOG EQU $      LOG
0265 009C 0000      DATA DC7331      UNIT 0
0266      009E' UTCST EQU $      CASSETTE
0267 009E 009C'      DATA DC7331      UNIT 0
0268 00A0 009E'      DATA DC7331      UNIT 1
0269      00A2' UTPTP EQU $      TAPE PUNCH
0270 00A2 0000      DATA DCTTYT
0271      00A4' UTPTR EQU $      TAPE READER
0272 00A4 00A2'      DATA DCTTYT
0273      *
0274      0009 RDACOD EQU 9      READ ASCII IO OP CODE
0275      0006 FSPCOD EQU 6      FORWARD SPACE IO OP CODE
0276      000F MXIOOP EQU 15    MAXIMUM IO OP

```

```

0278      00A6' IO      EQU $
0279 00A6 05CA      INCT R10      SET PRB ADDRESS
0280 00A8 C08B      MOV R11, R2
0281 00AA C0CA      MOV R10, R3
0282 00AC 58E0      SZCB @UEFMSK, @SYSFLG(R3)
      00AE 0097'
      00B0 0002
0283 00B2 D2A3      MOVVB @LUND(R3), R10      SET LUND
      00B4 0001
0284 00B6 098A      SRL R10, 8
0285 00B8 06A0      BL @CLDT
      00BA 0080'
0286 00BC 0989      SRL R9, 8      (DEVICE TYPE)
0287 00BE 0988      SRL R8, 8      (UNIT NUMBER)
0288 00C0 D1E3      MOVVB @I0OP(R3), R7      (IO OP CODE)
      00C2 0000
0289 00C4 0987      SRL R7, 8
0290 00C6 0289      CI R9, DUM
      00C8 007F
0291 00CA 160C      JNE I0020
0292 00CC 0287      CI R7, RDACOD      IF NOT READ OR FORWARD SPACE
      00CE 0009
0293 00D0 1303      JEQ LL2
0294 00D2 0287      CI R7, FSPCOD
      00D4 0006
0295 00D6 1605      JNE I0010
0296 00D8 00D8' LL2 EQU $
0297 00DA F8E0      SOCB @EFMASK, @SYSFLG(R3)
      00DC 0002
0298 00DE 04E3      CLR @CCOUNT(R3)
      00E0 0008
0299 00E2' I0010 EQU $
0300 00E2 0452      B *R2      RETURN
0301 00E4' I0020 EQU $
0302 00E4 0287      CI R7, MXI0OP
      00E6 000F
0303 00E8 1201      JLE LL4
0304 00EA 0452      B *R2
0305 00EC' LL4 EQU $
0306 00EC D1A9      MOVVB @OCTDT(R9), R6      (DISP TO BEGINNING OF TABLE)
      00EE 0098'
0307 00F0 0986      SRL R6, 8
0308 00F2 C148      MOV R8, R5      (UNIT*2)
0309 00F4 0A15      SLA R5, 1
0310 00F6 A185      A R5, R6
0311 00F8 C1A6      MOV @OCTDT(R6), R6
      00FA 0098'
0312 00FC D167      MOVVB @I0OPTBL(R7), R5      DISP TO DEV TYPE TABLE BY OPC
      00FE 0126'
0313 0100 1311      JEQ I0050      IF NULL ON IO OP CODE
0314 0102 0985      SRL R5, 8
0315 0104 C109      MOV R9, R4
0316 0106 0A14      SLA R4, 1      (DEV TYPE * 2)
0317 0108 A144      A R4, R5
0318 010A C165      MOV @I0OPTBL(R5), R5      ADDRESS OF CALLED SUBROUTINE
      010C 0126'
0319 010E 130A      JEQ I0050      IF NULL
0320 0110 0420      BLWP @ACL
      0112 0000

```

0321	0114	C283	MOV	R3, R10	MOVE PRB POINTER TO NEW WKSP
0322	0116	C246	MOV	R6, R9	SET CRU BASE ADDRESS
0323	0118	C329	MOV	@OCTCRU(R9), R12	
	011A	0000			
0324	011C	C045	MOV	R5, R1	CALL SUBROUTINE
0325	011E	0691	BL	*R1	
0326	0120	0420	BLWP	@RR	
	0122	0000			
0327		0124	I0050	EQU	\$
0328	0124	0452	B	*R2	
0329		0126	IOPTBL	EQU	\$
0330	0126	10	BYTE	OPEN-IOPTBL	00: OPEN
0331	0127	18	BYTE	CLOSE-IOPTBL	01: CLOSE
0332	0128	20	BYTE	CLEOF-IOPTBL	02: CLOSE EOF
0333	0129	28	BYTE	OPNREW-IOPTBL	03: OPEN REWIND
0334	012A	30	BYTE	CLUNL-IOPTBL	04: CLOSE UNLOAD
0335	012B	00	BYTE	0	
0336	012C	38	BYTE	FSPACE-IOPTBL	06: FORWARD SPACE
0337	012D	40	BYTE	BSPACE-IOPTBL	07: BACK SPACE
0338	012E	00	BYTE	0	
0339	012F	50	BYTE	RASCII-IOPTBL	09: READ ASCII
0340	0130	48	BYTE	RDIRCT-IOPTBL	010: READ DIRECT
0341	0131	60	BYTE	WASCII-IOPTBL	11: WRITE ASCII
0342	0132	58	BYTE	WDIRCT-IOPTBL	12: WRITE DIRECT
0343	0133	68	BYTE	WEOF-IOPTBL	13: WRITE END-/F-FILE
0344	0134	70	BYTE	REWIND-IOPTBL	14: REWIND
0345	0135	78	BYTE	UNLOAD-IOPTBL	15: UNLOAD
0346		0136	OPEN	EQU	\$
0347	0136	0000	DATA	0	
0348	0138	0000	DATA	CSOPEN	
0349	013A	0000	DATA	PTOPEN	
0350	013C	0000	DATA	PROPEN	
0351		013E	CLOSE	EQU	\$
0352	013E	0000	DATA	0	
0353	0140	0000	DATA	0	
0354	0142	0000	DATA	PTCLOS	
0355	0144	0000	DATA	PRCLOS	
0356		0146	CLEOF	EQU	\$
0357	0146	0000	DATA	0	
0358	0148	0000	DATA	0	
0359	014A	0000	DATA	PTCEOF	
0360	014C	0000	DATA	PTRILL	
0361		014E	OPNREW	EQU	\$
0362	014E	0000	DATA	0	
0363	0150	0000	DATA	0	
0364	0152	0000	DATA	PTOPRW	
0365	0154	013C	DATA	PROPEN	
0366		0156	CLUNL	EQU	\$
0367	0156	0000	DATA	0	
0368	0158	0000	DATA	0	
0369	015A	0142	DATA	PTCLOS	
0370	015C	0144	DATA	PRCLOS	
0371		015E	FSPACE	EQU	\$
0372	015E	0000	DATA	0	
0373	0160	0000	DATA	CSFWD	
0374	0162	0000	DATA	0	
0375	0164	0000	DATA	0	
0376		0166	BSPACE	EQU	\$
0377	0166	0000	DATA	0	
0378	0168	0000	DATA	CSBACK	

0379 016A 0000 DATA 0
0380 016C 0000 DATA 0
0381 016E EVEN
0382 016E' RDIRCT EQU \$
0383 016E 0000 DATA 0
0384 0170 0000 DATA 0
0385 0172 014C' DATA PTRILL
0386 0174 0000 DATA PRDIRT
0387 0176' RASCII EQU \$
0388 0176 0000 DATA LGRASC
0389 0178 0000 DATA CSRASC
0390 017A 0172' DATA PTRILL
0391 017C 0000 DATA PRDASC
0392 017E' WDIRCT EQU \$
0393 017E 0000 DATA 0
0394 0180 0000 DATA 0
0395 0182 0000 DATA PTDIRT
0396 0184 017A' DATA PTRILL
0397 0186' WASCII EQU \$
0398 0186 0000 DATA LGWASC
0399 0188 0000 DATA CSWASC
0400 018A 0000 DATA PTWASC
0401 018C 0184' DATA PTRILL
0402 018E' WEOF EQU \$
0403 018E 0000 DATA 0
0404 0190 0000 DATA CSWEOF
0405 0192 0000 DATA PTWEOF
0406 0194 0000 DATA 0
0407 0196' REWIND EQU \$
0408 0196 0000 DATA 0
0409 0198 0000 DATA CSRWND
0410 019A 0000 DATA 0
0411 019C 0000 DATA 0
0412 019E' UNLOAD EQU \$
0413 019E 0000 DATA 0
0414 01A0 0000 DATA CSUNLD
0415 01A2 0000 DATA 0
0416 01A4 0000 DATA 0
0417 END

NO ERRORS

TTYPT SDSMAC 947075 *D
PAPER TAPE SER. ROUTINE

13:15:10 FRIDAY, FEB 18, 1977.
948941-9901 **

PAGE 0001A

```
0004 IDT 'TTYPT'  
0005 *  
0006 * TITLE: TTYPTU  
0007 *  
0008 * DEVICE SERVICE ROUTINE - ASR 33 TTY AS PAPER  
0009 * TAPE UNIT  
0010 * ORIGINAL 9/5/76  
0011 * REVISION:  
0012 * COMPUTER: 990  
0013 * ABSTRACT:  
0014 * A GROUPING OF SUBROUTINES WHICH CONSTITUTE  
0015 * A ASR 33 TTY PAPER TAPE DEVICE SERVICE  
0016 * ROUTINE.  
0017 *  
0018 * TABLE OF CONTENTS  
0019 * 1. SUBROUTINES PUNCH  
0020 * A. OPEN (PTOPEN)  
0021 * B. CLOSE (PTCLOS)  
0022 * C. CLOSE EOF (PTCEOF)  
0023 * D. OPEN REWIND (PTOPRW)  
0024 * E. CLOSE UNLOAD (PTCLOS)  
0025 * F. WRITE ASCII (PTASC)  
0026 * G. WRITE DIRECT (PTDIRT)  
0027 * H. WRITE EOF (PTWEOF)  
0028 * 2. SUBROUTINE READER  
0029 * A. OPEN (PROPEN)  
0030 * B. CLOSE (PRCLOS)  
0031 * C. OPEN REWIND (PROPEN)  
0032 * D. CLOSE UNLOAD (PRCLOS)  
0033 * E. READ ASCII (PTRASC)  
0034 * F. READ DIRECT (RDIRCT)
```

```

0035      *
0036      *      PRB EQUATES DISPLACEMENTS
0037      *
0038      0000 IOOP EQU 0      IO OP CODE
0039      0001 LUNO EQU 1      LOGICAL UNIT NUMBER
0040      0002 SYSFLG EQU 2     SYSTEM FLAGS
0041      0003 USEFLG EQU 3     USER FLAGS
0042      0004 BFADDR EQU 4     BUFFER ADDRESS
0043      0006 BFLNTH EQU 6     BUFFER LENGTH
0044      0008 CCOUNT EQU 8     CHARACTER COUNT
0045      *
0046      *      FLAG SHIFT COUNTS
0047      *
0048      000F UE EQU 16-1      UNRECOVERABLE ERROR
0049      000D CHRMOD EQU 16-3  CHARACTER MODE
0050      000C NWFLG EQU 16-4   NO WAIT FLAG
0051      *
0052      *      REFERENCES
0053      *
0054      REF UEMASK      UNRECOVERABLE ERROR
0055      REF EFMASK
0056      *
0057      *      DEFINITIONS
0058      *
0059      DEF DCTTYT
0060      DEF PTOPEM
0061      DEF PTCLOS
0062      DEF PTCEOF
0063      DEF PTOPRW
0064      DEF PTDIRT
0065      DEF PTWASC
0066      DEF PTWEOF
0067      DEF PROPEN
0068      DEF PRCLOS
0069      DEF PRDIRT
0070      DEF PRDASC
0071      DEF PTRILL

```



```

0073      *
0074      *      CRU INTERFACE DEFINITIONS
0075      *
0076      000A RTS      EQU  >A      REQUEST TO SEND
0077      000B WRQ     EQU  >B      WRITE REQUEST
0078      000C RRQ     EQU  >C      READ REQUEST
0079      000D NSF     EQU  >D      NEW STATUS FLAG
0080      *
0081      *      WORKSPACE REGISTER DEFINITIONS
0082      *
0083      0000 R0      EQU  0
0084      0001 R1      EQU  1
0085      0002 R2      EQU  2
0086      0003 R3      EQU  3
0087      0004 R4      EQU  4
0088      0005 R5      EQU  5
0089      0006 R6      EQU  6
0090      0007 R7      EQU  7
0091      0008 R8      EQU  8
0092      0009 R9      EQU  9
0093      000A R10     EQU  10
0094      000B R11     EQU  11
0095      000C R12     EQU  12
0096      000D R13     EQU  13
0097      000E R14     EQU  14
0098      000F R15     EQU  15
0099      *
0100      *
0101      *      PROGRAM EQUATES
0102      *
0103      0000 ZERO    EQU  0      NULL VALUE
0104      0011 DC1     EQU  >11    READER-ON
0105      0012 DC2     EQU  >12    PUNCH-ON
0106      0013 DC3     EQU  >13    READER-OFF
0107      0014 DC4     EQU  >14    PUNCH-OFF
0108      0017 ETB     EQU  >17    CR REPLACEMENT CHAR. (CR TO ET
0109      0009 HT      EQU  9      HT VALUE
0110      000C FF      EQU  12     FORM FEED VALUE
0111      0007 BEL     EQU  7      BELL VALUE
0112      0008 BS      EQU  8      BACKSPACE VALUE
0113      000D CR      EQU  >0D    CARRIAGE RETURN VALUE
0114      000A LF      EQU  >0A    LINE FEED VALUE
0115      0020 SP      EQU  >20    SPACE VALUE
0116      007F DEL     EQU  >7F    BELETE VALUE
0117      0100 LBYTE   EQU  >100   LEFT BYTE POS. VALUE
0118      *
0119      *      INFORMATION DATA
0120      *
0121      0000 00      BYTE ZERO    FILLER
0122      0001 13     EOF          BYTE DC3      E-O-F VALUES
0123      0002 000A   EOR         DATA CR*LBYTE+LF  E-O-R VALUE
0124      0004 1300   DATA DC3*LBYTE+ZERO
0125      0006 0000   DATA ZERO*LBYTE+ZERO
0126      0008 00     BYTE ZERO
0127      000A 1400   EOR1        DATA DC4*LBYTE+ZERO
0128      000C 1200   EOR3        DATA DC2*LBYTE+ZERO
0129      000E 1100   DC1VAL     DATA DC1*LBYTE    DC1 LEFT BYTE
0130      0010 1300   DC3VAL     DATA DC3*LBYTE    DC3 LEFT BYTE
0131      0012 1700   ETBVAL     DATA ETB*LBYTE   ETB LEFT BYTE
0132      0014 0700   BELVAL     DATA BEL*LBYTE   BEL LEFT BYTE
  
```

TTYPT

SDSMAC 947075 *D

13:15:10 FRIDAY, FEB 18, 1977.

PAPER TAPE SER. ROUTINE

948941-9901 **

PAGE 0005

0133 0016 0000 CRVAL DATA CR*LBYTE
0134 0018 2000 ASCLOW DATA SP*LBYTE
0135 001A 7F00 ASCHIG DATA DEL*LBYTE
0136 001C 02 ERILOP BYTE 2
0137 001D 05 ERWRFG BYTE 5
0138 001E 0000 WRTFLG DATA 0
0139 0020 0000 REDFLG DATA 0
0140 0022 EVEN
0141 0022 0020 DCTTYT DATA >20

CR LEFT BYTE
LEGAL LOW ASCII VALUE
LEGAL HIGH ASCII VALUE+1
ILLEGAL OP CODE
DEVICE NOT OPEN

CRU BASE ADDR TTY

```

0143      *
0144      *      OPEN PAPER TAPE PUNCH
0145      *      (1) NAME: PTOPEM; SETS WRITE FLAG
0146      *
0147      *      CALL: BELOUT
0148      *
0149      *      (2) NAME: PTOPRW; SETS WRITE FLAG AND PUNCHES
0150      *      80 BLANK FRAMES
0151      *
0152      *      CALLS: PTOPEM AND PUNC80
0153      *
0154      0024' PTOPEM EQU $
0155      0024 C10B      MOV R11, R4      SAVE RETURN
0156      0026 0760      ABS @WRTFLG      IS DEVICE ALREADY OPEN?
0157      0028 001E'
0157      002A 166A      JNE OPEERR      YES, IN USE
0158      002C 05A0      INC @WRTFLG      SET WRITE FLAG
0159      002E 001E'
0159      0030 06A0      BL @BELOUT      OUTPUT TWO BELLS
0160      0032 0244'
0160      0034 0454      B *R4          RETURN
0161      *
0162      0036' PTOPRW EQU $
0163      0036 C08B      MOV R11, R2      SAVE RETURN
0164      0038 06A0      BL @PTOPEM
0165      003A 0024'
0165      003C 06A0      BL @PUNC80      PUNCH 80 BLANK FRAMES
0166      003E 0230'
0166      0040 0452      B *R2          RETURN
0167      *
0168      *      CLOSE PAPER TAPE PUNCH
0169      *      (1) NAME: PTCLOS; RESETS WRITE FLAG
0170      *
0171      *      (2) NAME: PTCEOF; RESETS WRITE FLAG, PUNCHES E0
0172      *      AND 80 BLANK FRAMES
0173      *
0174      *      CALLS: PUNC80 AND PUNE0F
0175      *
0176      0042' PTCLOS EQU $
0177      0042 C08B      MOV R11, R2      SAVE RETURN
0178      0044 04E0      CLR @WRTFLG      RESET WRITE FLAG
0179      0046 001E'
0179      0048 0452      B *R2          RETURN
0180      004A' PTCEOF EQU $
0181      004A C10B      MOV R11, R4      SAVE RETURN
0182      004C C08B      MOV R11, R2      SAVE RETURN FOR ERR
0183      004E 0760      ABS @WRTFLG      FLAG SET?
0184      0050 001E'
0184      0052 1356      JEQ OPEERR      NO, ERR
0185      0054 06A0      BL @PTWEOF
0186      0056 00EE'
0186      0058 04E0      CLR @WRTFLG      RESET WRITE FLAG
0187      005A 001E'
0187      005C 06A0      BL @PUNC80      PUNCH 80 BLANK FRAMES
0188      005E 0230'
0188      0060 0454      B *R4          RETURN
  
```

```

0190      *
0191      *      WRITE ASCII
0192      *      NAME: PTWASC; THIS ROUTINE GETS A CHAR. FROM
0193      *      USER'S BUFFER AND PUNCHES RESULT
0194      *      ON TAPE TERMINATES IF THE USER'S
0195      *      IS EMPTY.
0196      *      ERRORS: IF DEVICE IS NOT OPEN
0197      *
0198      0062' PTWASC EQU $
0199 0062 C08B      MOV R11, R2      SAVE RETURN
0200 0064 04C8      CLR R8          CLEAR OFF SET
0201 0066 0760      ABS @WRTFLG     IS DEVICE OPEN?
      0068 001E'
0202 006A 134A      JEQ OPEERR     NO, RETURN
0203 006C 8A88      NXTCHR C      R8, @CCOUNT(R10)  BUFFER EMPTY?
      006E 0008
0204 0070 1318      JEQ EOREXT     RETURN
0205 0072 06A0      BL @GETCHR     GET NEXT CHAR.
      0074 0270'
0206 0076 0249      ANDI R9, DEL*LBYTE  NO PARITY, JUST CHAR.
      0078 7F00
0207 007A 0289      CI R9, CR*LBYTE   IS IT CR?
      007C 0D00
0208 007E 1603      JNE LEGCHK     NO, IS VALUE LEGAL?
0209 0080 D260      MOVB @ETBVAL, R9  CR TO ETB
      0082 0012'
0210 0084 1003      JMP ASCPUN     DEPOSIT RESULTS
0211 0086 06A0      LEGCHK BL @LEGCHR  IS CHAR LEGAL?
      0088 01F0'
0212 008A 1002      JMP NEXINC     OFF SET INCREASE (ERR RT)
0213 008C 06A0      ASCPUN BL @PUNOUT  PUNCH ONE CHAR. (GOOD RT)
      008E 0286'
0214 0090 0588      NEXINC INC R8    INCREASE OFF SET
0215 0092 0289      CI R9, DC4*LBYTE  IS IT PUNCH-OFF?
      0094 1400
0216 0096 16EA      JNE NXTCHR     NO, NEXT CHAR
0217 0098 0209      LI R9, DC2*LBYTE  YES, PUNCH-ON CHAR
      009A 1200
0218 009C 06A0      BL @PUNOUT     TURN PUNCH BACK ON
      009E 0286'
0219 00A0 10E5      JMP NXTCHR     CONTINUE
0220      *
0221      *      WRITE END OF RECORD OR END OF FILE
0222      *
0223      00A2' EOREXT EQU $
0224 00A2 0208      LI R8, 7        NUMB. CHAR. TO BE PUNCHED
      00A4 0007
0225 00A6 0207      LI R7, EOR      CHAR. ADDR. TO PUNCH
      00A8 0002'
0226 00AA D277      EOROUT MOVB *R7+, R9  CHAR. VALUE TO PUNCH
0227 00AC 06A0      BL @PUNOUT     PUNCH CHAR.
      00AE 0286'
0228      DEC R8          DONE?
0229 00B2 16FB      JNE EOROUT     NO, CONTINUE
0230 00B4 0452      B *R2          RETURN

```

```

0232      *
0233      *      WRITE DIRECT
0234      *      NAME: WDIRCT; THIS ROUTINE WRITES THE SPECIFIED
0235      *      # OF CHARS. AND TERMINATES WHEN
0236      *      CHAR. COUNT IS DONE, CHECKS
0237      *      WRITE FLAG.
0238      *      CALLS: WEOF
0239      *
0240      00B6' PTDIRT EQU $
0241      00B6 04C8      CLR R8      CLEAR OFF SET
0242      00B8 C08B      MOV R11, R2  SAVE RETURN
0243      00BA 0760      ABS @WRTFLG  WRITE FLAG SET?
0244      00BC 001E'
0244      00BE 1320      JEQ OPEERR  NO, DEVICE NOT OPEN
0245      00C0 8A88      WDRLOP C  R8, @CCOUNT(R10)  OUTPUT DONE?
0246      00C2 0008
0246      00C4 130F      JEQ WDREND  GO TO EOF SET UP
0247      00C6 06A0      BL @GETCHR  GET CHAR.
0247      00C8 0270'
0248      00CA 0249      ANDI R9, >FF00  JUST CHAR.
0248      00CC FF00
0249      00CE 06A0      BL @PUNOUT  PUNCH CHAR.
0249      00D0 0286'
0250      00D2 0289      CI R9, DC4*LBYTE  PUNCH OFF?
0250      00D4 1400
0251      00D6 1604      JNE NOTDC4  NO, CONTINUE
0252      00D8 0209      LI R9, DC2*LBYTE  TURN PUNCH-ON
0252      00DA 1200
0253      00DC 06A0      BL @PUNOUT  DO IT
0253      00DE 0286'
0254      00E0 0588      NOTDC4 INC R8      INCREASE OFF SET
0255      00E2 10EE      JMP WDRLOP  CONTINUE
0256      00E4 0208      WDREND LI R8, 4  # CHAR TO PUNCH
0256      00E6 0004
0257      00E8 0207      LI R7, EOR+2  CHAR ADDR. TO PUNCH
0257      00EA 0004'
0258      00EC 10DE      JMP EOROUT  GO DO IT
0259      *
0260      *      WRITE END OF FILE
0261      *      NAME: PTWEOF; THIS ROUTINE PUNCHES AN E/O/F
0262      *      FOR EITHER DIRECT OR ASCII CHECK
0263      *      SEE IF DEVICE IS OPEN
0264      *
0265      00EE' PTWEOF EQU $
0266      00EE C08B      MOV R11, R2  SAVE RETURN
0267      00F0 0760      ABS @WRTFLG  FLAG SET?
0267      00F2 001E'
0268      00F4 1305      JEQ OPEERR  NO, DEVICE NOT OPEN
0269      00F6 0208      LI R8, 8  PUNCH 8 SEQ. CHAR.
0269      00F8 0008
0270      00FA 0207      LI R7, EOF  CHAR. ADDR. TO PUNCH
0270      00FC 0001'
0271      00FE 10D5      JMP EOROUT  OUTPUT END OF FILE
  
```

```

0273          *
0274          *   ERROR RETURN ROUTINES
0275          *
0276          0100' OPEERR EQU $
0277 0100 FAA0      SOCB @UEMASK,@SYSFLG(R10) UNRECOV. ERR
          0102 0000
          0104 0002
0278 0106 DAA0      MOVB @ERWRFG,@USEFLG(R10) DEVICE NOT OPEN
          0108 001D'
          010A 0003
0279 010C 0452      B *R2          RETURN
0280 010E          PTRILL EVEN
0281 010E FAA0      SOCB @UEMASK,@SYSFLG(R10) UNRECOV. ERR
          0110 0102'
          0112 0002
0282 0114 DAA0      MOVB @ERILOP,@USEFLG(R10) ILLEGAL OP CODE
          0116 001C'
          0118 0003
0283 011A 045B      RT          RETURN
0284          *
0285          *   OPEN PAPER TAPE READER
0286          *   NAME: PROPEN; THIS ROUTINE SETS READER FLAG
0287          *
0288          011C' PROPEN EQU $
0289 011C 0760      ABS @REDFLG      IS DEVICE ALREADY OPEN?
          011E 0020'
0290 0120 16EF      JNE OPEERR      YES, IN USE
0291 0122 05A0      INC @REDFLG      SET READ FLAG
          0124 0020'
0292 0126 045B      RT          RETURN
0293          *
0294          *   CLOSE PAPER TAPE READER
0295          *   NAME: PRCL0S; THIS ROUTINE RESETS READER FLAG
0296          *
0297          0128' PRCL0S EQU $
0298 0128 04E0      CLR @REDFLG      RESET READ FLAG
          012A 0020'
0299 012C 045B      RT          RETURN

```

```

0301      *
0302      *      READ ASCII
0303      *      NAME: PTRASC; THIS ROUTINE READS CHAR. FROM THE
0304      *      TAPE CHECKS FOR LEGAL CHAR. AND TERMINATE
0305      *      IF USER'S BUFFER IS FULL OR CR IS
0306      *      ENCOUNTERED, IGNORES LEADING BLANKS.
0307      *      ERRORS: IF DEVICE NOT OPEN
0308      *
0309      012E' PRDASC EQU $
0310      012E C08B      MOV R11, R2      SAVE RETURN
0311      0130 04C8      CLR R8          CLEAR OFF SET
0312      0132 0760      ABS @REDFLG     DEVICE OPEN?
           0134 0020'
0313      0136 13E4      JEQ OPEERR     NO, ERR RETURN
0314      0138 C260      MOV DC1VAL, R9   TURN ON READER
           013A 000E'
0315      013C 06A0      BL @PUNOUT     SEND TO TTY
           013E 0286'
0316      0140 06A0      NULoop BL @REDCHR    GET CHAR.
           0142 027A'
0317      0144 0249      ANDI R9, DEL*LBYTE  REMOVE PARITY, JUST CHAR.
           0146 7F00
0318      0148 0289      CI R9, ZERO*LBYTE  IS IT NULL?
           014A 0000
0319      014C 13F9      JEQ NULoop     YES, IGNORE
0320      014E 0289      FIRCHR CI R9, DC3*LBYTE  IS IT END OF FILE?
           0150 1300
0321      0152 1327      JEQ FILEND    YES, RETURN
0322      0154 8A88      CHRLOP C R9, @BFLNTH(R10)  BUFFER FULL?
           0156 0006
0323      0158 131B      JEQ BUFEND    YES, LOOK FOR EOR
0324      015A 0289      CI R9, CR*LBYTE  IS IT A CR?
           015C 0D00
0325      015E 1318      JEQ BUFEND    YES, TERMINATE
0326      0160 0289      CI R9, ETB*LBYTE  IS IT A ETB?
           0162 1700
0327      0164 1603      JNE CHKLEG    NO, IS IT STILL LEGAL?
0328      0166 D260      MOV @CRVAL, R9  CHANGE ETB TO CR
           0168 0016'
0329      016A 1003      JMP CHRSTR     GO TO STORE PROCESS
0330      016C 06A0      CHKLEG BL @LEGCHR  CHECK FOR LEGAL CHAR.
           016E 01F0'
0331      0170 1009      JMP OFFINC     OFF SET INCREASE
0332      0172 06A0      CHRSTR BL @STRCHR  STORE RESULTS
           0174 0266'
0333      0176 0289      CI R9, DC3VAL  IS IT READER-OFF?
           0178 0010'
0334      017A 1604      JNE OFFINC     NO, NEXT CHAR
0335      017C 0209      LI R9, DC1VAL  YES, READER-ON VALUE
           017E 000E'
0336      0180 06A0      BL @PUNOUT     TURN READER BACK ON
           0182 0286'
0337      0184 0588      OFFINC INC R8   INCREASE OFF SET
0338      0186 06A0      BL @REDCHR     GET NEXT CHAR.
           0188 027A'
0339      018A 0249      ANDI R9, DEL*LBYTE  REMOVE PARITY
           018C 7F00
0340      018E 10E2      JMP CHRLOP     TEST CHAR. LOOP
0341      0190 06A0      BUFEND BL @EORCHR  MOVE TAPE TO EOR
           0192 0220'

```

```

0342 0194 CA88          MOV  R8,@CCOUNT(R10)  STORE CHARS. READ
      0196 0008
0343 0198 C260  PRDOFF MOV  @DC3VAL,R9  TURN READER OFF
      019A 0010'
0344 019C 06A0          BL   @PUNOUT          SEND TO TTY
      019E 0286'
0345 01A0 0452          B    *R2              NOT E-O-F RETURN
0346 01A2 FAA0  FILEND SOCB @EFMASK,@SYSFLG(R10) SET EOF FLAG
      01A4 0000
      01A6 0002
0347 01A8 10F7          JMP  PRDOFF          RETURN
0348
0349                    *
0350                    *   READ DIRECT
0351                    *   NAME: RDIRCT; THIS ROUTINE READS ALL CHARS. AND
0352                    *   TERMINATES IF THE BUFFER GETS FULL, IGNOR
0353                    *   LEADING NULLS
0354                    *
0354      01AA' PRDIRT EQU  $
0355 01AA 04C8          CLR  R8              CLEAR OFF SET
0356 01AC C08B          MOV  R11,R2          SAVE RETURN
0357 01AE 0760          ABS  @REDFLG        IS DEVICE OPEN?
      01B0 0020'
0358 01B2 13A6          JEQ  OPEERR        DEVICE NOT OPEN
0359 01B4 C260          MOV  DC1VAL,R9     TURN READER ON
      01B6 000E'
0360 01B8 06A0          BL   @PUNOUT        SEND TO TTY
      01BA 0286'
0361 01BC 06A0  NULLOP BL   @REDCHR        GET CHAR.
      01BE 027A'
0362 01C0 0249          ANDI R9,>FF00      CHAR. + PARITY
      01C2 FF00
0363 01C4 0289          CI   R9,ZERO*LBYTE IS IT NULL?
      01C6 0000
0364 01C8 13F9          JEQ  NULLOP        YES, IGNORE
0365 01CA 8A88  NTDIRC C   R8,@BFLNTH(R10) BUFFER FULL?
      01CC 0006
0366 01CE 130F          JEQ  RDIRDN        RETURN
0367 01D0 06A0          BL   @STRCHR       STORE CHAR.
      01D2 0266'
0368 01D4 0289          CI   R9,DC3VAL    IS IT READER-OFF,
      01D6 0010'
0369 01D8 1604          JNE  UPINC         NO, NEXT CHAR
0370 01DA 0209          LI   R9,DC1VAL    YES, READER-ON VALUE
      01DC 000E'
0371 01DE 06A0          BL   @PUNOUT        TURN READER BACK ON
      01E0 0286'
0372 01E2 0588  UPINC  INC  R8              INCREASE OFF SET
0373 01E4 06A0          BL   @REDCHR       GET NEXT CHAR.
      01E6 027A'
0374 01E8 0249          ANDI R9,>FF00      CHAR. + PARITY
      01EA FF00
0375 01EC 10EE          JMP  NTDIRC        CONTINUE
0376 01EE 10D0  RDIRDN JMP  BUFEND       LOOK FOR END-OF-RECORD

```



```

0378          * CHECKS FOR LEGAL ASCII CHARACTER
0379          01F0' LEGCHR EQU $
0380 01F0 C0CB      MOV R11, R3      SAVE RETURN
0381 01F2 0289      CI R9, HT*LBYTE  IS IT HT?
          01F4 0900
0382 01F6 1312      JEQ LEGEND      YES, LEGAL CHAR.
0383 01F8 0289      CI R9, FF*LBYTE  IS IT FF?
          01FA 0C00
0384 01FC 130F      JEQ LEGEND      YES, LEGAL CHAR.
0385 01FE 0289      CI R9, BEL*LBYTE IS IT BEL?
          0200 0700
0386 0202 130C      JEQ LEGEND      YES, LEGAL CHAR.
0387 0204 0289      CI R9, BS*LBYTE  IS IT BS?
          0206 0800
0388 0208 1309      JEQ LEGEND      YES, LEGAL CHAR.
0389 020A 0289      CI R9, DC3VAL    DC3 FOR STANDALONE
          020C 0010'
0390 020E 1306      JEQ LEGEND      YES, LEGAL CHAR.
0391 0210 9809      CB R9, @ASCLOW  BELOW BOTTOM RANGE?
          0212 0018'
0392 0214 1104      JLT LEGBAD      NOT LEGAL CHAR
0393 0216 9809      CB R9, @ASCHIG  ABOVE TOP RANGE?
          0218 001A'
0394 021A 1401      JHE LEGBAD      NOT LEGAL CHAR
0395 021C 05C3      LEGEND INCT R3  PROPER RETURN
0396 021E 0453      LEGBAD B *R3    RETURN
0397          *
0398          * LOOKS FOR READER OFF CHARACTER
0399          *
0400          0220' EORCHR EQU $
0401 0220 C0CB      MOV R11, R3      SAVE RETURN
0402 0222 0289      ROFLOP CI R9, DC3*LBYTE  READER OFF?
          0224 1300
0403 0226 1303      JEQ EOREND      YES, QUIT
0404 0228 06A0      BL @REDCHR      GET NEXT CHAR.
          022A 027A'
0405 022C 10FA      JMP ROFLOP
0406 022E 0453      EOREND B *R3    NO, KEEP LOOKING
          RETURN
  
```

```

0408      *
0409      * PUNCHES 80 BLANK FRAMES
0410      *
0411      0230' PUNC80 EQU $
0412 0230 C0CB      MOV R11, R3      SAVE RETURN
0413 0232 0208      LI R8, 80      CHAR. TO PUNCH
      0234 0050
0414 0236 0209 PUNLOP LI R9, ZERO      NULL IN R9
      0238 0000
0415 023A 06A0      BL @PUNOUT      PUNCH CHAR.
      023C 0286'
0416 023E 0608      DEC R8      DONE?
0417 0240 16FA      JNE PUNLOP      NO, CONTINUE
0418 0242 0453      B *R3      RETURN
0419      *
0420      *
0421      * OUTPUTS TWO BELLS WAITS FOR CR
0422      *
0423      0244' BELOUT EQU $
0424 0244 C0CB      MOV R11, R3      SAVE RETURN
0425 0246 C260      MOV @EOR1, R9      TURN PUNCH OFF
      0248 000A'
0426 024A 06A0      BL @PUNOUT      PUNCH IT
      024C 0286'
0427 024E C260      MOV @BELVAL, R9      PUT BELL VALUE IN R9
      0250 0014'
0428 0252 06A0      BL @PUNOUT      PUNCH IT
      0254 0286'
0429 0256 06A0      BL @PUNOUT      PUNCH IT AGAIN
      0258 0286'
0430 025A 06A0 DELINP BL @REDCHR      GET CHAR.
      025C 027A'
0431 025E 0289      CI R9, DEL*LBYTE      IS IT A DEL?
      0260 7F00
0432 0262 16FB      JNE DELINP      NO, KEEP WAITING
0433 0264 0453      B *R3      RETURN

```

```

0435      *
0436      *      PUTS CHARACTER INTO USER'S BUFFER
0437      *
0438      0266' STRCHR EQU $
0439 0266 C06A      MOV  @BFADDR(R10),R1      GET ADDRESS
      0268 0004
0440 026A A048      A      R8,R1      BUFFER OFF SET
0441 026C D449      MOV#  R9,*R1      STORE
0442 026E 045B      RT      RETURN
0443      *
0444      *      GETS CHARACTER FROM USER'S BUFFER
0445      *
0446      0270' GETCHR EQU $
0447 0270 C06A      MOV  @BFADDR(R10),R1      GET ADDRESS
      0272 0004
0448 0274 A048      A      R8,R1      BUFFER OFFSET
0449 0276 D251      MOV#  *R1,R9      GET VALUE
0450 0278 045B      RT      RETURN
0451      *
0452      *      READS ONE CHARACTER
0453      *
0454 027A      REDCHR EVEN
0455 027A 1F0C      REDRDY TB  RRQ      READY TO READ
0456 027C 16FE      JNE  REDRDY      NO, WAIT
0457 027E 04C9      CLR  R9      I/O STORAGE REGISTER CLEAR
0458 0280 3609      STCR R9,8      PUT IN LEFT BYTE
0459 0282 1D0C      SBO  RRQ      CLEAR REQUEST
0460 0284 045B      RT      RETURN
0461      *
0462      *      WRITE ONE CHARACTER
0463      *
0464 0286      PUNOUT EVEN
0465 0286 1D0A      SBO  RTS      TERMINAL READY?
0466 0288 3209      LDCR R9,8      GET VALUE
0467 028A 1F0B      WRQRDY TB  WRQ      CLEAR IT
0468 028C 16FE      JNE  WRQRDY      NO, WAIT
0469 028E 1D0B      SBO  WRQ      NOW CLEAR IT
0470 0290 1E0A      SBZ  RTS      RESET
0471 0292 045B      RT      RETURN
0472      END

```

NO ERRORS

TTYPT LABEL

SDSMAC 947075 *D 13:15:10 FRIDAY, FEB 18, 1977.

VALUE DEFN REFERENCES 948941-9901 **

PAGE 0016

			0050	0076	0077	0078	0079	0083	0084	0085	0086
			0087	0088	0089	0090	0091	0092	0093	0094	0095
			0096	0097	0098	0103	0104	0105	0106	0107	0108
			0109	0110	0111	0112	0113	0114	0115	0116	0117
			0154	0162	0176	0180	0198	0223	0240	0265	0276
			0288	0297	0309	0354	0379	0400	0411	0423	0438
			0446								
ERILOP	R	001C	0136	0282							
ERWRFG	R	001D	0137	0278							
ETB		0017	0108	0131	0326						
ETBVAL	R	0012	0131	0209							
EVEN				0140	0280	0454	0464				
FF		000C	0110	0383							
FILEND	R	01A2	0346	0321							
FIRCHR	R	014E	0320								
GETCHR	R	0270	0446	0205	0247						
HT		0009	0109	0381							
IDT				0004							
INC				0158	0214	0254	0291	0337	0372		
INCT				0395							
IOOP		0000	0038								
JEQ				0184	0202	0204	0244	0246	0268	0313	0319
				0323	0325	0358	0364	0366	0382	0384	0386
				0390	0403						
JHE				0394							
JLT				0392							
JMP				0210	0212	0219	0255	0258	0271	0329	0331
				0347	0375	0376	0405				
JNE				0157	0208	0216	0229	0251	0290	0327	0334
				0417	0432	0456	0468				
LBYTE		0100	0117	0123	0124	0125	0127	0128	0129	0130	0131
				0133	0134	0135	0206	0207	0215	0217	0250
				0317	0318	0320	0324	0326	0339	0363	0381
				0385	0387	0402	0431				
LDCR				0466							
LEGBAD	R	021E	0396	0392	0394						
LEGCHK	R	0086	0211	0208							
LEGCHR	R	01F0	0379	0211	0330						
LEGEND	R	021C	0395	0382	0384	0386	0388	0390			
LF		000A	0114	0123							
LI				0217	0224	0225	0252	0256	0257	0269	0270
				0370	0413	0414					
LUNO		0001	0039								
MOV				0155	0163	0177	0181	0182	0199	0242	0266
				0314	0342	0343	0356	0359	0380	0401	0412
				0425	0427	0439	0447				
MOVb				0209	0226	0278	0282	0328	0441	0449	
NEXINC	R	0090	0214	0212							
NOTDC4	R	00E0	0254	0251							
NSF		000D	0079								
NTDIRC	R	01CA	0365	0375							
NULLOP	R	01BC	0361	0364							
NULLOOP	R	0140	0316	0319							
NWFLG		000C	0050								
NXTCHR	R	006C	0203	0216	0219						
OFFINC	R	0184	0337	0331	0334						
OPEERR	R	0100	0276	0157	0184	0202	0244	0268	0290	0313	0358
PAGE				0002	0003	0034	0072	0142	0189	0231	0272
				0377	0407	0434					0300
PRCLOS	D	0128	0297	0068							


```

0003 * PROCEDURE RUN(CPL);
0004 * /* RUN CONTROLS THE EXECUTION OF
0005 * A USER PROGRAM UNDER EITHER SIE
0006 * OR TRACE MODE. IT PROVIDES THE
0007 * USER'S KEYBOARD LEVEL CONTROL
0008 * OF THE SYSTEM.
0009 * */
0010 * IF CPL.PARM1 .NE. NULL THEN DO;
0011 * INSTRUCTION_COUNT = CPL.PARM1;
0012 * END;
0013 * DO UNTIL RETURN_TO_OPERATOR;
0014 * CALL SCAN_TABLE(TRACE, USER_PC, FOUND);
0015 * IF .NOT. FOUND THEN DO;
0016 * CALL SIE(USER_PC, USER_WP, USER_ST);
0017 * END; ELSE DO;
0018 * IF TRACE NE RESIDENT THEN ERROR EXIT;
0019 * CALL TRACER(USER_PC, USER_WP, USER_ST);
0020 * END;
0021 * IF WRITE_PROTECT_ERROR = 1 THEN ERROR EXIT
0022 * IF USER_PC < LOW_LIMIT .OR. USER_PC >
0023 * HIGH_LIMIT THEN DO;
0024 * INSTRUCTION_COUNT = INSTRUCTION_COUNT -1;
0025 * CALL SCAN_TABLE(BREAKPOINT, USER_PC, FOUND);
0026 * IF FOUND THEN DO;
0027 * CALL BREAKPOINT_PROCESS(USER_PC, BREAK_#,
0028 * BREAK_ENTRY; TERM);
0029 * IF TERM THEN SIGNAL RETURN_TO_OPERATOR
0030 * END;
0031 * IF INSTRUCTION_COUNT .EQ. 0 THEN
0032 * SIGNAL RETURN_TO_OPERATOR;
0033 * IF INSTRUCTION_COUNT .EQ. -1 THEN
0034 * INSTRUCTION_COUNT = INSTRUCTION_COUNT +1;
0035 * CALL MONITOR_CHAR(CHAR, NO_WAIT);
0036 * IF CHAR .EQ. ESCAPE THEN SIGNAL
0037 * RETURN_TO_OPERATOR;
0038 * END;
0039 * END;
0040 * RETURN_TO_OPERATOR:
0041 * END RUN;
0042 * IDT: RUNPGN
0043 * TITLE: RUN
0044 * RUN USER PGM UNDER DEBUG
0045 * REVISION:
0046 * ORIGINAL
0047 * 12/13/76 MODIFIED FROM ORIGINAL PROTOTYPING
0048 * SYSTEM TO CORRECT A RUN COUNT COMPUTA
0049 * ERROR.
0050 * COMPUTER: 990, ASM
0051 * ABSTRACT: THIS ROUTINE CONTROLS THE DEBUG EXECUTION
0052 * OF A USER'S PROGRAM.
0053 * CALLING SEQUENCE:
0054 * BL @RUN
0055 * R10 = PTR TO COMMOND PARAMETER LIST
0056 *
0057 * REF'S AND DEF'S
0058 *
0059 * DEF RUN
0060 * REF GETBUF
0061 * REF RETBUF
  
```

```

0063 REF RWP
0064 REF ACL
0065 REF RR
0066 REF INSCNT
0067 REF TRACE
0068 REF ERROR
0069 REF BKPT
0070 REF BKPTPR
0071 REF PCOUT
0072 REF LOWLIM
0073 REF HILIM
0074 REF ESC
0075 REF SIE
0076 REF TRACER
0077 REF SREGN
0078 REF MONCHR
0079 REF USRPC
0080 REF SCTB
0081 REF STRPRT
0082 REF CRUPRT

```

INSTRUCTION COUNT

```

*
*WORKSPACE REGISTER DEFINITIONS

```

```

0085 *
0086 0000 R0 EQU 0
0087 0001 R1 EQU 1
0088 0002 R2 EQU 2
0089 0003 R3 EQU 3
0090 0004 R4 EQU 4
0091 0005 R5 EQU 5
0092 0006 R6 EQU 6
0093 0007 R7 EQU 7
0094 0008 R8 EQU 8
0095 0009 R9 EQU 9
0096 000A R10 EQU 10
0097 000B R11 EQU 11
0098 000C R12 EQU 12
0099 000D R13 EQU 13
0100 000E R14 EQU 14
0101 000F R15 EQU 15

```

```

0102 *
0103 *
0104 0004 MX04 EQU >0004
0105 0007 MX07 EQU >0007
0106 0001 PRTVIO EQU 1

```

```

TRY TO EX TRACE WHEN NOT IN O
WRITE PROTECT VIOLATION

```

```

0107 *
0108 *
0109 0000' RUN EQU #
0110 * *ALLOC, COPY, LINK
0111 0000 0420 BLWP @ACL
0112 0002 0000

```

1-PROCEDURE RUN(CPL);

```

0112 *
0113 *
0114 *
0115 *
0116 *
0117 *
0118 *
0119 0004 C27A NOV *R10+, R9
0120 0006 0A99 SLA R9, 9

```

```

2-/* RUN CONTROLS THE EXECUTIO
2- A USER PROGRAM UNDER EITH
2- OR TRACE MODE. IT PROVIDE
2- USER'S KEYBOARD LEVEL CON
2- OF THE SYSTEM.
2-*/
2-IF CPL.PARM1 .NE. NULL THEN

```

```
0122 * 3-INSTRUCTION_COUNT = CPL.PARM
0123 000A C81A MOV *R10,@INSCNT
000C 0000
0124 * 2-END;
0125 000E' RUN010 EQU $
0126 * 2-DO UNITL RETURN_TO_OPERATOR;
0127 * 3-CALL SCAN_TABLE<TRACE,USER_P
0128 000E 020A LI R10,TRACE
0010 0000
0129 0012 C260 MOV @USRPC,R9
0014 0000
0130 0016 06A0 BL @SCTB
0018 0000
0131 * 3-IF .NOT. FOUND THEN DO;
0132 001A C28A MOV R10,R10 0 => FOUND
0133 001C 1303 JEQ RUN020
0134 001E' RUN015 EQU $
0135 * 4-CALL SIE<USER_PC,USER_WP,USE
0136 001E 06A0 BL @SIE
0020 0000
0137 * 3-END; ELSE DO;
0138 0022 1008 JMP RUN030
0139 0024' RUN020 EQU $
0140 *****CHECK THAT TRACE OVERLAY IS PRESENT
0141 * 4-IF TRACE NE RESIDENT THEN ER
0142 0024 C2A0 MOV @SREGN,R10
0026 0000
0143 0028 132C JEQ RUN055
0144 * 4-CALL TRACER<USER_PC,USER_WP,
0145 002A 06A0 BL @TRACER
002C 0000
0146 * 3-END;
0147 002E 102D JMP RUN060
0148 0030 06A0 BL @PCOUT
0032 0000 OUTPUT USRPC TO PANEL
0149 0034' RUN030 EQU $
0150 * 3-IF WRITE_PROTECT_ERROR = 1 T
0151 0034 020A LI R10,MX07
0036 0007
0152 0038 020C LI R12,CRUPRT
003A 0000
0153 003C 1F01 TB PRTV10
0154 003E 1E03 JNE RUN032
0155 0040 3020 LDCR @STRPRT,0 RESET V10 FLAG AND RESTORE PR
0042 0000
0156 0044 1020 JMP RUN057 ERROR EXIT
0157 0046' RUN032 EQU $
0158 0046 C260' MOV @USRPC,R9
0048 0014'
0159 * 3-IF USER_PC < LOWLIMIT .OR.
0160 * 4-HIGHLIMIT THEN DO;
0161 004A 8809 C R9,@LOWLIM
004C 0000
0162 004E 1A03 JL RUN035
0163 0050 8809 C R9,@HILIM
0052 0000
0164 0054 1ADC JL RUN010
0165 0056' RUN035 EQU $
0166 * 4-INSTRUCTION_COUNT = INSTRUC
```

```

0058 000C'
0168 005A 1303      JEQ  RUN040      IF COUNT=0, SKIP DEPLETION
0169 005C 0620      DEC  @INSCNT      DECREMENT COUNT.
      005E 0058'
0170 0060 1314      JEQ  RUN060      IF DEPLETED, RETURN TO MONI
0171          *      4-CALL SCAN_TABLE<BREAKPOINT,U
0172 0062 020A      RUN040 LI  R10,BKPT
      0064 0000
0173 0066 06A0      BL   @SCTB
      0068 0018'
0174          *      4-IF FOUND THEN DO;
0175 006A C28A      MOV  R10,R10
0176 006C 1604      JNE  RUN050
0177          *      5-CALL BREAKPOINT_PROCESS<USER
0178          *      6-BREAK_ENTRY; TERM);
0179 006E 06A0      BL   @BKPTPR
      0070 0000
0180          *      4-IF TERM THEN SIGNAL RETURN_T
0181 0072 C28A      MOV  R10,R10
0182 0074 160A      JNE  RUN060
0183          *      4-END;
0184          0076' RUN050 EQU  $
0185          *      4-CALL MONITOR_CHAR<CHAR, NO_WA
0186 0076 06A0      BL   @MONCHR
      0078 0000
0187          *      4-IF CHAR .EQ. ESCAPE THEN SIG
0188          *      5-RETURN_TO_OPERATOR;
0189 007A 9809      CB   R9,@ESC
      007C 0000
0190 007E 16C7      JNE  RUN010
0191 0080 1004      JMP  RUN060
0192          *      3-END;
0193          *      2-END;
0194          0082' RUN055 EQU  $
0195 0082 020A      LI   R10,MX04
      0084 0004
0196          0086' RUN057 EQU  $
0197 0086 06A0      BL   @ERROR
      0088 0000
0198          *      2-RETURN_TO_OPERATOR;
0199          008A' RUN060 EQU  $
0200          *      1-END RUN;
0201          *      *LINK TO PREV WKSP, RET CURR WKSP
0202 008A 0420      BLWP @RR
      008C 0000
0203 008E 045B      RT
0204          END

```

NO ERRORS

LABEL VALUE DEFN REFERENCES 936187-9901 **

RR	E	008C	0065	0202				
RT				0203				
RUN	D	0000	0109	0059				
RUN010	R	000E	0125	0121	0164	0190		
RUN015	R	001E	0134					
RUN020	R	0024	0139	0133				
RUN030	R	0034	0149	0138				
RUN032	R	0046	0157	0154				
RUN035	R	0056	0165	0162				
RUN040	R	0062	0172	0168				
RUN050	R	0076	0184	0176				
RUN055	R	0082	0194	0143				
RUN057	R	0086	0196	0156				
RUN060	R	008A	0199	0147	0170	0182	0191	
RWP	E		0063					
SCTB	E	0063	0080	0130	0173			
SIE	E	0020	0075	0136				
SLA				0120				
SREGN	E	0026	0077	0142				
STRPRT	E	0042	0081	0155				
TB				0153				
TITL				0001				
TRACE	E	0010	0067	0128				
TRACER	E	002C	0076	0145				
USRPC	E	0048	0079	0129	0158			


```
0002          IDT  'BASCON'
0003          *
0004          *   THIS PROGRAM TAKES AN ASCII NUMBER AND
0005          *   CONVERTS IT TO A SPECIFIC BASE IN 2'S
0006          *   COMPLEMENT NOTATION.  THE POSITIVE
0007          *   NUMBER MUST BE IN 7-BIT ASCII FORMAT
0008          *   STARTING AT LOCATION (INDEPENDANT OF PROGRAM
0009          *   BASE ADDRESS) >104, ONE DIGIT PER BYTE.
0010          *   THE BASE VALUE SHOULD BE A DATA WORD AT
0011          *   LOCATION >100; THE LENGTH OF THE NUMBER
0012          *   TO BE CONVERTED, IN DIGITS, SHOULD BE AT
0013          *   DATA LOCATION >102.
0014          *   THE RESULT IS RETURNED AT LOCATION >120.
0015          *   NO ASCII VALIDITY CHECKS OR OVERFLOW CHECKS
0016          *   ARE MADE.
0017          *
0018 0000 02E0  ENTRY  LWPI  WRKS
          0002 0122
0019 0004 04C2          CLR   R2          R2=LOOP COUNTER
          0006 04C3          CLR   R3          R3=INTERMEDIATE ACCUMULATOR
0021 0008 38E0  LOOP   MPY   @BASE,R3      MULT. SUBRESULT BY BASE
          000A 0100
0022 000C C0C4          MOV   R4,R3          LOAD LEAST SIG. BITS OF RESULT
0023 000E D122          MOVB @NUMBER(R2),R4
          0010 0104
0024 0012 0984          SRL   R4,8
0025 0014 A0C4          A     R4,R3          ADD NEXT DIGIT
0026 0016 0223          AI    R3,-30
          0018 FFE2
0027 001A 0582          INC   R2
0028 001C 8802          C     R2,@LENGTH
          001E 0102
0029 0020 16F3          JNE   LOOP          JUMP BACK IF NOT DONE
0030 0022 C803  MOV   MOV   R3,@RESULT
          0024 0120
0031 0026 2FE0          XOP  @OUT,15
          0028 0142
0032 0100          AORG >100
0033 0100 0010  BASE   DATA >10
0034 0102 0006  LENGTH DATA >6
0035 0104          NUMBER BSS  >10
0036 0120          AORG >120
0037 0120 0000  RESULT DATA 0
0038 0122          WRKS  BSS  >20
0039 0142 0400  OUT    DATA >0400
0040 0144 0000'  END   ENTRY
```

NO ERRORS

PROGRAM LABEL	VALUE	DEFN	REFERENCES
A			0025
AI			0026
AORG			0032 0036
BASE	0100	0033	0021
BSS			0035 0038
C			0028
CLR			0019 0020
DATA			0033 0034 0037 0039
END			0040
ENTRY R	0000	0018	0040
IDT			0002
INC			0027
JNE			0029
LENGTH	0102	0034	0028
LOOP R	0008	0021	0029
LWFI			0018
MOV R	0022	0030	0022 0030
MOVB			0023
MPY			0021
NUMBER	0104	0035	0023
OUT	0142	0039	0031
R2	0002		0019 0023 0027 0028
R3	0003		0020 0021 0022 0025 0026 0030
R4	0004		0022 0023 0024 0025
RESULT	0120	0037	0030
SRL			0024
TITL			0001
WRKS	0122	0038	0018
XOP			0031


```
0002          IDT 'OPTEST'
0003          *
0004          * TITLE:      OPTEST
0005          *             TELETYPE I/O OP-CODE TEST PROGRAM
0006          * REVISION:  NOV. 30, 1976
0007          *             ORIGINAL
0008          * COMPUTER:  990
0009          * ABSTRACT:
0010          *             TESTS THE VARIOUS INPUT/OUTPUT SUPERVISOR
0011          *             CALLS TO/FROM THE TELETYPE INTERFACE.
0012          *
0013          * REGISTER EQUATES
0014          *
0015          0000 R0      EQU 0
0016          0001 R1      EQU 1
0017          0002 R2      EQU 2
0018          0003 R3      EQU 3
0019          0004 R4      EQU 4
0020          000B R11     EQU 11
0021          *
0022 0000 WSP      BSS 32          WORK SPACE REGISTERS
0023 0020 02E0 START  LWPI WSP     GET WORK SPACE
0024          0022 0000'
0024 0024 2FE0 AGAIN  XOP @HEAD, 15  INPUT S, E, O FOE STOP, ERR PRO
0025          0026 00E8'
0025 0028 2FE0          XOP @REPLY, 15  OR OP CODE PRO
0026          002A 0154'
0026          *
0027          * DECISION BRANCH R2=LUNO; R3= OP CODE
0028          *
0029 002C C060          MOV @IMP, R1      GET INPUT
0029          002E 01DE'
0030 0030 0241          ANDI R1, >7F00    MASK OFF GARBAGE
0030          0032 7F00
0031 0034 0221          AI R1, ->4500    IS IT AN E?
0031          0036 B000
0032 0038 1308          JEQ ERRPRO      YES, ERROR PROCEDURE
0033 003A 0221          AI R1, ->A00     IS IT AN O?
0033          003C F600
0034 003E 1316          JEQ OPCPRO      YES, OP CODE PROCEDURE
0035 0040 0221          AI R1, ->400     IS IT AN S?
0035          0042 FC00
0036 0044 16EF          JNE AGAIN       NO, PROMP AGAIN
0037          *
0038          * RETURN TO MONITOR
0039          *
0040 0046 2FE0          XOP @RET, 15     RETURN
0040          0048 0152'
0041          *
0042          * ERROR PROCEDURE
0043          *
0044 004A ERRPRO EVEN
0045 004A 2FE0          XOP @ASKOP, 15    OUTPUT PROMPT
0045          004C 00F4'
0046 004E 2FE0          XOP @INPCD, 15    INPUT REPLY
0046          0050 0100'
0047 0052 D820          MOVB @INBUF, @VAL  SET UP FOR CONVERSION
0047          0054 01E0'
0047          0056 0111'
0048 0058 2FE0          XOP @COMBI, 15    CONVERT
```

```
005A 010C'
0049 005C 06C0      SWPB R0      RESULTS IN LEFT BYTE
0050 005E D800      MOVB R0,@I0OP  DEPOSIT FOR PROCESS
0060 0126'
0051 0062 06A0      BL @TESTOT    PROCESS CALL
0064 00C8'
0052 0066 04E0      CLR @SYSFL    CLEAR ALL ERR FLAGS
0068 0128'
0053 006A 10DC      OVER JMP AGAIN DO IT AGAIN
0054 *
0055 * OP CODE PROCEDURE
0056 *
0057 006C      OPCPRO EVEN
0058 006C 2FE0      XOP @ASKOP,15  OUTPUT PROMPT
006E 00F4'
0059 0070 2FE0      XOP @INPCD,15  GET REPLY
0072 0100'
0060 0074 D820      MOVB @INBUF,@VAL SET UP FOR CONVERSION
0076 01E0'
0078 0111'
0061 007A 2FE0      XOP @COMBI,15  CONVERT
007C 010C'
0062 007E 06C0      SWPB R0      PUT IN LEFT BYTE
0063 0080 C0C0      MOV R0,R3    GET REPLY VALUE
0064 0082 0243      ANDI R3,>7F00 FIRST REPLY ONLY
0084 7F00
0065 0086 C0A0      MOV @INBUF,R2
0088 01E0'
0066 008A 0242      ANDI R2,>7F   SECOND REPLY ONLY
008C 007F
0067 008E 0282      CI R2,>50     IS IT PUNCH?
0090 0050
0068 0092 1605      JNE REDTES   NO, CHECK ASCII OR DIRECT
0069 0094 D800      MOVB R0,@I0OP DEPOSIT RESULTS
0096 0126'
0070 0098 06A0      BL @TESTOT    PROCESS CALL
009A 00C8'
0071 009C 10C3      JMP AGAIN    DO IT AGAIN
0072 *
0073 * CHECK READ ASCII OR DIRECT
0074 *
0075 009E      REDTES EVEN
0076 009E 0282      CI R2,>52     IS IT READ?
00A0 0052
0077 00A2 16C0      JNE AGAIN    NO, BAD START
0078 00A4 0283      CI R3,>A00    IS IT READ DIRECT?
00A6 0A00
0079 00A8 1603      JNE WRTASC   CHECK READ ASCII
0080 00AA 2FE0      XOP @REDDIR,15 PROCESS IT
00AC 0146'
0081 00AE 10DD      JMP OVER     DO IT AGAIN
0082 00B0 0283      WRTASC CI R3,>900 IS IT READ ASCII?
00B2 0900
0083 00B4 1603      JNE REDOPN   IS IT OPEN READER?
0084 00B6 2FE0      XOP @REDASC,15 PROCESS IT
00B8 0130'
0085 00BA 10D7      JMP OVER     DO IT AGAIN
0086 *
0087 00BC 0283      REDOPN CI R3,>0 IS OP CODE OPEN VALUE?
00BE 0000
```

```

0088 00C0 16D4      JNE  OVER          NO, BAD START
0089 00C2 2FE0      XOP  @OPNRED, 15   YES, OPEN READER
      00C4 013C
0090 00C6 10D1      JMP  OVER          PROMPT AGAIN
0091
0092 *
0093 * ALL PURPOSE XOP
0094 00C8          TESTOT EVEN
0095 00C8 2FE0      XOP  @SUPCL, 15   PROCESS CALL
      00CA 0124
0096 00CC C820      MOV  @CARLIN, @OUTBUF  GET CR, LF
      00CE 01A8
      00D0 0114
0097 00D2 04E0      CLR  @OUT2
      00D4 0116
0098 00D6 2FE0      XOP  @OUTRES, 15   OUTPUT CR, LF
      00D8 0118
0099 00DA C020      MOV  @SYSFL, R0     FLAGS IN R0
      00DC 0128
0100 00DE 2FE0      XOP  @CONAS, 15    CONVERT TO ASCII
      00E0 0112
0101 00E2 2FE0      XOP  @OUTRES, 15   OUTPUT RESULTS
      00E4 0118
0102 00E6 045B      B    *R11          RETURN
0103
0104 *
0105 * PRB DATA TABLES
0106 00E8          HEAD   EVEN
0107 00E8 0000      DATA 0            OUTPUT MESS TO LOG
0108 00EA 0B00      DATA >B00
0109 00EC 0000      DATA 0
0110 00EE 0160      DATA HEADER
0111 00F0 0000      DATA 0
0112 00F2 004C      DATA 76
0113
0114 00F4 0000      ASKOP DATA 0       OUTPUT PROMPT MESS
0115 00F6 0B00      DATA >B00
0116 00F8 0000      DATA 0
0117 00FA 01AA      DATA OUTMES
0118 00FC 0000      DATA 0
0119 00FE 0034      DATA 52
0120
0121 0100 0000      INPCD DATA 0       REPLY
0122 0102 0900      DATA >900
0123 0104 0000      DATA 0
0124 0106 01E0      DATA INBUF
0125 0108 0002      DATA 2
0126 010A 0000      DATA 0
0127
0128 010C 0D00      COMBI DATA >0D00   CONVERT REPLY TO FORM
0129 010E 30        TEXT '000'
      010F 30
      0110 30
0130 0111 30        VAL   BYTE >30
0131
0132 0112          CONAS  EVEN
0133 0112 0C00      DATA >C00
0134 0114 0000      OUTBUF DATA 0
0135 0116 0000      OUT2  DATA 0
0136

```

```
0137 0118      OUTRES EVEN
0138 0118 0000      DATA 0
0139 011A 0B00      DATA >B00
0140 011C 0000      DATA 0
0141 011E 0114      DATA OUTBUF
0142 0120 0000      DATA 0
0143 0122 0004      DATA 4
0144      *
0145 0124 0000 SUPCL DATA 0      MAIN CALL TO R/P
0146 0126 00      IOOP  BYTE 0
0147 0127 0A      LUNO  BYTE >A
0148 0128 00      SYSFL BYTE 0
0149 0129 00      USERFL BYTE 0
0150 012A 01E2      BUFAD DATA BUFFER1
0151 012C 0000      BUFLIN DATA 0
0152 012E 0014      CHRCT DATA 20
0153      *
0154 0130      REDASC EVEN      READ ASCII
0155 0130 0000      DATA 0
0156 0132 0909      IOOPA DATA >909
0157 0134 0000      DATA 0
0158 0136 01F6      DATA BUFFER2
0159 0138 0014      DATA 20
0160 013A 0000      DATA 0
0161      *
0162 013C 0000      OPNRED DATA 0
0163 013E 0009      DATA >009      OPEN READER
0164 0140 0000      DATA 0, 0, 0
      0142 0000
      0144 0000
0165      *
0166 0146 0000      REDDIR DATA 0      READ DIRECT
0167 0148 0A09      IOOPD DATA >A09
0168 014A 0000      DATA 0
0169 014C 020A      DATA BUFFER3
0170 014E 0014      DATA 20
0171 0150 0000      DATA 0
0172      *
0173 0152      RET      EVEN
0174 0152 0400      DATA >400
0175      *
0176 0154 0000      REPLY DATA 0      BRANCH INPUT
0177 0156 0900      DATA >900
0178 0158 0000      DATA 0
0179 015A 01DE      DATA IMP
0180 015C 0001      DATA 1
0181 015E 0000      DATA 0
0182      *
0183      * PRB STORAGE OUTPUT AREA
0184      *
0185 0160 0A00      HEADER DATA >A00
0186 0162 49      TEXT 'INPUT S FOR STOP OR E FOR ERROR PROCEDURE'
      0163 4E
      0164 50
      0165 55
      0166 54
      0167 20
      0168 53
      0169 20
      016A 46
```

016B 4F
016C 52
016D 20
016E 53
016F 54
0170 4F
0171 50
0172 20
0173 4F
0174 52
0175 20
0176 45
0177 20
0178 46
0179 4F
017A 52
017B 20
017C 45
017D 52
017E 52
017F 4F
0180 52
0181 20
0182 50
0183 52
0184 4F
0185 43
0186 45
0187 44
0188 55
0189 52
018A 45
0187 018C 0A0D
0188 018E 4F
018F 52
0190 20
0191 4F
0192 20
0193 46
0194 4F
0195 52
0196 20
0197 4F
0198 50
0199 20
019A 43
019B 4F
019C 44
019D 45
019E 20
019F 50
01A0 52
01A1 4F
01A2 43
01A3 45
01A4 44
01A5 55
01A6 52
01A7 45
0189 01A8 0A0D

DATA >A0D
TEXT 'OR 0 FOR OP CODE PROCEDURE'

0189 01A8 0A0D CARLIN DATA >A0D

```
0190 *
0191 01AA 0A00 OUTMES DATA >A00
0192 01AC 45 TEXT 'ENTER DESIRED OP-CODE AND MODE (R=READ,
01AD 4E
01AE 54
01AF 45
01B0 52
01B1 20
01B2 44
01B3 45
01B4 53
01B5 49
01B6 52
01B7 45
01B8 44
01B9 20
01BA 4F
01BB 50
01BC 20
01BD 43
01BE 4F
01BF 44
01C0 45
01C1 20
01C2 41
01C3 4E
01C4 44
01C5 20
01C6 4D
01C7 4F
01C8 44
01C9 45
01CA 20
01CB 28
01CC 52
01CD 3D
01CE 52
01CF 45
01D0 41
01D1 44
01D2 2C
0193 01D3 20 TEXT ' P=PUNCH'
01D4 50
01D5 3D
01D6 50
01D7 55
01D8 4E
01D9 43
01DA 48
0194 01DC 0A00 DATA >A00
0195 *
0196 01DE 0000 IMP DATA 0
0197 01E0 0000 INBUF DATA 0
0198 01E2 20 BUFFR1 TEXT ' !0123ABCD'
01E3 21
01E4 30
01E5 31
01E6 32
01E7 33
01E8 41
```



```
01E9 42
01EA 43
01EB 44
0199 01EC 7E7F      DATA >7E7F, >0809, >070A, >0102, >2020
01EE 0809
01F0 070A
01F2 0102
01F4 2020
0200 01F6          BUFFR2 BSS 20
0201 020A          BUFFR3 BSS 20
0202 0020          END START
NO ERRORS
```


LABEL	VALUE	DEFN	REFERENCES	936146-9901	**									
REDDIR	R 0146	0166	0080											
REDOPN	R 00BC	0087	0083											
REDTES	R 009E	0075	0068											
REPLY	R 0154	0176	0025											
RET	R 0152	0173	0040											
START	R 0020	0023	0202											
SUPCL	R 0124	0145	0095											
SWPB			0049	0062										
SYSFL	R 0128	0148	0052	0099										
TESTOT	R 00C8	0094	0051	0070										
TEXT			0129	0186	0188	0192	0193	0198						
TITL			0001											
USERFL	R 0129	0149												
VAL	R 0111	0130	0047	0060										
WRTASC	R 00B0	0082	0079											
WSP	R 0000	0022	0023											
XOP			0024	0025	0040	0045	0046	0048	0058	0059	0061			
			0080	0084	0089	0095	0098	0100	0101					