



```

LL      AAAAAA  SSSSSSSS  WW      WW  EEEEEEEEE  EEEEEEEEE  PPPPPPP
LL      AAAAAA  SSSSSSSS  WW      WW  EEEEEEEEE  EEEEEEEEE  PPPPPPP
LL      AA      AA  SS      WW      WW  EE      EE      PP      PP
LL      AA      AA  SS      WW      WW  EE      EE      PP      PP
LL      AA      AA  SS      WW      WW  EE      EE      PP      PP
LL      AA      AA  SS      WW      WW  EE      EE      PP      PP
LL      AA      AA  SSSSSS  WW      WW  EEEEEEE  EEEEEEE  PPPPPPP
LL      AA      AA  SSSSSS  WW      WW  EEEEEEE  EEEEEEE  PPPPPPP
LL      AAAAAAAAAA  SS      WW  WW  WW  EE      EE      PP
LL      AAAAAAAAAA  SS      WW  WW  WW  EE      EE      PP
LL      AA      AA  SS      WWW  WWW  EE      EE      PP
LL      AA      AA  SS      WWW  WWW  EE      EE      PP
LL      AA      AA  SSSSSSS  WW      WW  EEEEEEE  EEEEEEE  PP
LL      AA      AA  SSSSSSS  WW      WW  EEEEEEE  EEEEEEE  PP
LLLLLLLLLLL  AA      AA  SSSSSSS  WW      WW  EEEEEEE  EEEEEEE  PP
LLLLLLLLLLL  AA      AA  SSSSSSS  WW      WW  EEEEEEE  EEEEEEE  PP

```

```

....
....
....
....

```

```

LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS

```

(2)	53	DECLARATIONS
(3)	77	LPA\$SETIBF - INITIALIZE IBUF ARRAY
(4)	175	START SWEEP ROUTINES
(5)	433	LPA\$STPSWP - STOP SWEEP
(6)	502	LPA\$CLOCKA - SET CLOCK A RATE
(7)	562	LPA\$CLOCKB - SET CLOCK B RATE
(8)	673	LPA\$LAMSKS - SET MASKS BUFFER
(9)	778	LPA\$SETADC - SET CHANNEL PARAMETERS
(10)	875	LPA\$CVADF - CONVERT A/D TO FLOATING POINT
(10)	876	LPA\$FLT16 - CONVERT UNSIGNED WORD TO FLOATING POINT
(11)	942	LPA\$XRATE - COMPUTE CLOCK RATE AND PRESET
(12)	1051	LPA\$LOADMC - LOAD MICROCODE
(13)	1148	LPA\$ASSIGN - ASSIGN A CHANNEL TO AN LPA-11

```
0000 1 .TITLE LPA$SWEEP
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 :*****
0000 6 :*
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :*
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :*
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27 :
0000 28 :
0000 29 :++
0000 30 : FACILITY: LPA-11 PROCEDURE LIBRARY
0000 31 :
0000 32 : ABSTRACT:
0000 33 : THIS MODULE CONTAINS THE START SWEEP ROUTINES FOR THE LPA-11
0000 34 : PROCEDURE LIBRARY.
0000 35 :
0000 36 : ENVIRONMENT: USER MODE, SHARED OR NON-SHARED LIBRARY
0000 37 :
0000 38 : AUTHOR: STEVE BECKHARDT, CREATION DATE: 23-AUG-78
0000 39 :
0000 40 : MODIFIED BY:
0000 41 :
0000 42 : V03-001 SBL3001 Steven B. Lionel 30-Mar-1982
0000 43 : Change module name to LPA$SWEEP.
0000 44 :
0000 45 : V04 SRB0001 STEVE BECKHARDT 23-OCT-1979
0000 46 : FIXED BUG IN HANDLING OF BUFFER OVERRUN NON-FATAL:
0000 47 : INITIALIZED BUFFER 0 RELEASED FLAG TO 1 IN ROUTINE
0000 48 : SWPCOM IF BUFFER OVERRUN IS NON-FATAL. AS A RESULT,
0000 49 : BUFFER 0 SHOULD NOT BE RELEASED THE FIRST TIME.
0000 50 :
0000 51 :--
```

DECLARATIONS

```
0000 53      .SBTTL  DECLARATIONS
0000 54      :
0000 55      : INCLUDE FILES:
0000 56      :
0000 57      :
0000 58      :
0000 59      : MACROS:
0000 60      :
0000 61      :
0000 62      :
0000 63      : EQUATED SYMBOLS:
0000 64      :
0000 65      :
0000 66      :
0000 67      : OWN STORAGE:
0000 68      :
0000 69      :
00000000 70      .PSECT  _LPA$CODE,PIC,SHR,EXE,NOWRT,BYTE
0000 71      :
0000 72      : PREFIX STRING USED IN ASSIGNING CHANNEL
0000 73      :
24 31 31 41 50 4C 0000 74 DNPREFIX:      .ASCII  /LPA11$/
00000006 0006 75 DNPREFIXS=.-DNPREFIX
```

LPASSETIBF - INITIALIZE IBUF ARRAY

```

0006 77 .SBTTL LPASSETIBF - INITIALIZE IBUF ARRAY
0006 78 :++
0006 79 : FUNCTIONAL DESCRIPTION:
0006 80 :
0006 81 : THIS ROUTINE INITIALIZES THE IBUF ARRAY
0006 82 :
0006 83 : CALLING SEQUENCE:
0006 84 :
0006 85 : CALLS/G
0006 86 :
0006 87 : INPUT PARAMETERS:
0006 88 :
0006 89 : IBUF(AP) ADDRESS OF IBUF ARRAY (MUST BE LONGWORD ALIGNED)
0006 90 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
0006 91 : LAMSKB(AP) ADDRESS OF LAMSKS ARPAY
0006 92 : BUFO(AP) ADDRESS OF FIRST DATA BUFFER
0006 93 :
0006 94 :
0006 95 : BUFN(AP) ADDRESS OF LAST DATA BUFFER (UP TO 8 ARE ALLOWED)
0006 96 :
0006 97 : IMPLICIT INPUTS:
0006 98 :
0006 99 : NONE
0006 100 :
0006 101 : OUTPUT PARAMETERS:
0006 102 :
0006 103 : IND(AP) ; ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
0006 104 :
0006 105 : IMPLICIT OUTPUTS:
0006 106 :
0006 107 : NONE
0006 108 :
0006 109 : COMPLETION CODES:
0006 110 :
0006 111 : 1 INDICATES IBUF WAS SUCCESSFULLY INITIALIZED
0006 112 : 0 INDICATES AN ARGUMENT ERROR - POSSIBLE CAUSES:
0006 113 : 1) INCORRECT NUMBER OF ARGUMENTS
0006 114 : 2) IBUF ARRAY NOT LONGWORD ALIGNED
0006 115 : 3) BUFFER ADDRESSES NOT EQUIDISTANT
0006 116 :
0006 117 : SIDE EFFECTS:
0006 118 :
0006 119 : NONE
0006 120 :
0006 121 : --
0006 122 :
007C 0C06 123 .ENTRY LPASSETIBF,^M<R2,R3,R4,R5,R6>
0008 124
0008 125 CLRL R2 ; RETURN STATUS
56 04 AC D0 000A 126 MOVL 4(AP),R6 ; GET ADDRESS OF IBUF ARRAY
56 03 D3 000E 127 BITL #3,R6 ; IS IBUF LONGWORD ALIGNED?
66 00A8 8F 00 66 00 7C 0011 128 BNEQ 60$ ; NO - ERROR
54 0C AC DE 001B 129 MOVCS #0,(R6),#0,#IBF$K_LENGTH,(R6) ; ZERO IBUF
001F 130 MOVAL 12(AP),R4 ; POINT TO LAMSKB ARG
001F 131
001F 132 ; PROCESS LAMSKB ARG
1C A6 84 D0 001F 133 MOVL (R4)+,IBF$L_LAMSKB(R6) ; STORE IN IBUF

```

LPASSETIBF - INITIALIZE IBUF ARRAY

				0023	134				
				0023	135				
	53	6C	9A	0023	136				
	53	04	C2	0026	137				
	07	53	D1	0029	138				
		54	1A	002C	139				
	2C	A6	64	D0	002E	140			
	22	A6	53	90	0032	141			
			14	13	0036	142			
14	A6	64	84	C3	0038	143			
			0A	11	003D	144			
					003F	145			
	50	64	84	C3	003F	146	30\$:		
	14	A6	50	D1	0043	147			
			39	12	0047	148			
		F3	53	F5	0049	149	40\$:		
					004C	150			
					004C	151	50\$:		
50	A6	50	A6	DE	004C	152			
54	A6	50	A6	DE	0051	153			
					0056	154			
58	A6	58	A6	DE	0056	155			
5C	A6	58	A6	DE	005B	156			
					0060	157			
60	A6	60	A6	DE	0060	158			
64	A6	60	A6	DE	0065	159			
					006A	160			
24	A6	48	A6	3E	006A	161			
					006F	162			
38	A6	00010001	01000001	8F	006F	163			
					007B	164			
					007B	165			
	66	1234	8F	B0	007B	166			
					0080	167			
			52	D6	0080	168			
					0082	169			
	51	08	AC	D0	0082	170	60\$:		
			03	13	0086	171			
	61	52	D0	0088	172				
			04	008B	173	70\$:			

  

```

; PROCESS BUFFER ADDRESSES
MOVZBL (AP),R3 ; GET NUMBER OF ARGS
SUBL #4,R3 ; ACCOUNT FOR FIRST ARGUMENTS
CPL R3,#7 ; R3 = # OF BUFFERS - 1
BGTRU 60$ ; INCORRECT # OF ARGS
MOVL (R4),IBF$L_CMDTBL+CMT$L_BFRADDR(R6); STORE FIRST BUFFER ADDR.
MOVB R3,IBF$L_CMDTBL+CMT$L_VBFRMASK(R6) ; STORE # IN COMMAND TABLE
BEQL 50$ ; ONLY 1 BUFFER ADDRESS SPECIFIED
SUBL3 (R4)+,(R4),IBF$L_LBUF(R6) ; COMPUTE AND STORE BUFFER LENGTH
BRB 40$

30$: SUBL3 (R4)+,(R4),R0 ; COMPUTE NEXT LENGTH
CPL R0,IBF$L_LBUF(R6) ; MAKE SURE IT AGREES
BNEQ 60$ ; IT DOESN'T - ERROR
SOBGR R3,30$ ; DO NEXT ONE

50$: ; INITIALIZE ARGUMENT INDEPENDENT STUFF. FIRST INIT. QUEUES
MOVAL IBF$L_USRQFL(R6),IBF$L_USRQFL(R6) ; USER QUEUE
MOVAL IBF$L_USRQFL(R6),IBF$L_USRQBL(R6)

MOVAL IBF$L_DEVQFL(R6),IBF$L_DEVQFL(R6) ; DEVICE QUEUE
MOVAL IBF$L_DEVQFL(R6),IBF$L_DEVQBL(R6)

MOVAL IBF$L_INUQFL(R6),IBF$L_INUQFL(R6) ; IN USE QUEUE
MOVAL IBF$L_INUQFL(R6),IBF$L_INUQBL(R6)

MOVAV IBF$L_USW(R6),IBF$L_CMDTBL+CMT$L_USWADDR(R6) ; USW ADDRESS
MOVQ #*X1000101000001,IBF$L_CMDTBL+CMT$L_DELAY(R6); DEFAULT SAMPLING
; NUMBERS

MOVW #INITCODE,IBF$L_IOST(R6) ; SHOW THAT SETIBF WAS CALLED

INCL R2 ; INDICATE SUCCESS

60$: MOVL 8(AP),R1 ; GET ADDRESS OF IND
BEQL 70$ ; DEFAULTED
MOVL R2,(R1) ; RETURN STATUS
RET
    
```

START SWEEP ROUTINES

```

008C 175 .SBTTL START SWEEP ROUTINES
008C 176 :++
008C 177 : FUNCTIONAL DESCRIPTION:
008C 178 :
008C 179 : THESE ROUTINES ARE THE START SWEEP ROUTINES. THERE ARE
008C 180 : FOUR TYPES OF SWEEPS: A/D, D/A, DIGITAL IN, AND DIGITAL OUT.
008C 181 :
008C 182 : CALLING SEQUENCE:
008C 183 :
008C 184 : CALLS/G
008C 185 :
008C 186 : INPUT PARAMETERS:
008C 187 :
008C 188 : IBUF(AP) ADDRESS OF ARRAY INITIALIZED BY SETIBF
008C 189 : LBUF(AP) ADDRESS OF WORD CONTAINING THE SIZE OF EACH DATA BUFFER
008C 190 : (IN WORDS)
008C 191 : NBUF(AP) ADDRESS OF LONGWORD CONTAINING NUMBER OF BUFFERS TO FILL
008C 192 : MODE(AP) ADDRESS OF A WORD WHICH SPECIFIES SAMPLING OPTIONS
008C 193 : BIT 5 SERIAL/PARALLEL (DUAL A/D)
008C 194 : BIT 6 DEDICATED/MULTIREQUEST MODE
008C 195 : BIT 9 CLOCK OVERFLOW/EXTERNAL TRIGGER
008C 196 : BIT 10 TIME STAMPING
008C 197 : BIT 11 EVENT MARKING
008C 198 : BIT 12 IMMEDIATE START/DIGITAL INPUT START
008C 199 : BIT 13 SINGLE/DUAL A/D
008C 200 : BIT 14 BUFFER OVER/UNDERRUN FATAL/NON-FATAL
008C 201 : DWELL(AP) ADDRESS OF WORD CONTAINING DWELL VALUE
008C 202 : IEFN(AP) VALUE OF EVENT FLAG OR IF GREATER THAN 128 ADDRESS OF
008C 203 : COMPLETION ROUTINE
008C 204 : LDELAY(AP) ADDRESS OF A WORD CONTAINING DELAY VALUE
008C 205 : ICHN(AP) ADDRESS OF A BYTE CONTAINING START CHANNEL VALUE
008C 206 : NCHN(AP) ADDRESS OF A WORD CONTAINING NUMBER OF SAMPLES VALUE
008C 207 : IND(AP) ADDRESS OF A LONGWORD TO RECEIVE STATUS
008C 208 :
008C 209 : IMPLICIT INPUTS:
008C 210 :
008C 211 : VARIOUS FIELDS IN THE IBUF ARRAY
008C 212 :
008C 213 : OUTPUT PARAMETERS:
008C 214 :
008C 215 : IND(AP) ADDRESS OF A LONGWORD TO RECEIVE STATUS
008C 216 :
008C 217 : IMPLICIT OUTPUTS:
008C 218 :
008C 219 : VARIOUS FIELDS IN THE IBUF ARRAY
008C 220 :
008C 221 : COMPLETION CODES:
008C 222 :
008C 223 : 1 INDICATES SUCCESS
008C 224 : 0 INDICATES ERROR DETECTED BY THIS ROUTINE - POSSIBILITIES:
008C 225 : 1) SETIBF WAS NOT CALLED FIRST
008C 226 : 2) RLSBUF HAS NOT BEEN CALLED TO RELEASE A BUFFER
008C 227 : 3) SIZE OF DATA BUFFERS DISAGREES WITH SIZE
008C 228 : COMPUTED FROM SETIBF CALL.
008C 229 : VARIOUS VMS CODES RETURNED BY $ASSIGN AND $QIO
008C 230 :
008C 231 : SIDE EFFECTS:

```



START SWEEP ROUTINES

```

008C 232 :
008C 233 :
008C 234 :
008C 235 :--
008C 236 :
55 02 00FC 008C 237 .ENTRY LPASADSWP,^M<R2,R3,R4,R5,R6,R7> ; START A/D SWEEP
    17 11 00BE 238 MOVL #2,R5 ; MODE WORD
    0091 239 BRB SWPCOM
    0093 240
    0093 241
55 0082 8F 00FC 0093 242 .ENTRY LPASDASWP,^M<R2,R3,R4,R5,R6,R7> ; START D/A SWEEP
    OE 11 0095 243 MOVZWL #^X82,R5 ; MODE WORD
    009A 244 BRB SWPCOM
    009C 245
    009C 246
55 1A 00FC 009C 247 .ENTRY LPASDISWP,^M<R2,R3,R4,R5,R6,R7> ; START DIG. INPUT SWEEP
    07 11 009E 248 MOVL #^X1A,R5 ; MODE WORD
    00A1 249 BRB SWPCOM
    00A3 250
    00A3 251
55 009A 8F 00FC 00A3 252 .ENTRY LPASDOSWP,^M<R2,R3,R4,R5,R6,R7> ; START DIG. OUT SWEEP
    3C 3C 00A5 253 MOVZWL #^X9A,R5
    00AA 254
    00AA 255
    00AA 256
57 04 AC DE 00AA 257 SWPCOM: ; COMMON PROCESSING FOR ALL SWEEP ROUTINES. R5 CONTAINS MODE WORD.
53 6C 02 83 00AE 258 MOVL 4(AP),R7 ; POINT TO FIRST ARGUMENT
    50 D4 00B2 261 SUBB3 #2,(AP),R3 ; R3 CONTAINS # OF OPTIONAL ARGS
    00B2 260 CLRL R0 ; ASSUME ERROR
    00B4 262
    1234 56 87 D0 00B4 263 MOVL (R7)+,R6 ; ADDRESS OF IBUF ARRAY
    8F 66 B1 00B7 264 CMPW IBF$Q_IOST(R6),#INITCODE ; VERIFY SETIBF WAS CALLED
    18 12 00BC 265 BNEQ 15$ ; IT WASN'T - ERROR
    66 7C 00BE 266 CLRQ IBF$Q_IOST(R6) ; CLEAR USER'S I/O STATUS BLOCK
    00C0 267
    11 4C A6 00 00C0 268 ; VERIFY THAT A BUFFER HAS BEEN RELEASED AND SET IN USW
    E1 00C0 269 BBC #FLG_V_USWSET,IBF$W_FLAGS(R6),15$ ; BR. IF USW IS NOT SET
    00C5 270
    00C5 271 ; PROCESS LBUF
    51 97 3C 00C5 272 MOVZWL @ (R7)+,R1 ; GET LBUF
    51 02 C4 00C8 273 MULL #2,R1 ; MULTIPLY BY 2 TO CONV. WORDS TO BYTES
    52 14 A6 D0 00CB 274 MCVL IBF$L_LBUF(R6),R2 ; GET LBUF CALCULATED IN SETIBF
    08 13 00CF 275 BEQL 20$ ; THERE ISN'T ONE
    52 51 D1 00D1 276 CMPL R1,R2 ; COMPARE THEM
    03 13 00D4 277 BEQL 20$ ; EQUAL - NO ERROR
    00D6 278
    0160 31 00D6 279 15$: BRW 115$ ; ERROR
    00D9 280
    14 A6 51 D0 00D9 281 20$: MOVL R1,IBF$L_LBUF(R6) ; STORE LENGTH OF EACH BUFFER
    52 22 A6 9A 00DD 282 MOVZBL IBF$L_CMDTBL+CMT$B_VBFMASK(R6),R2 ; GET # OF BUFFERS-1
    52 D6 00E1 283 INCL R2 ; ADD 1
    28 A6 51 52 C5 00E3 284 MULL3 R2,R1,IBF$L_CMDTBL+CMT$L_BFRLEN(R6) ; STORE OVERALL BFR LENGTH
    00E8 285
    00E8 286 ; PROCESS NBUF
    53 97 00E8 287 DECB R3 ; DECR. ARG COUNT
    OF 19 00EA 288 BLSS 30$ ; ARG OMITTED

```

START SWEEP ROUTINES

51	87	D0	00EC	289	MOVL	(R7)+,R1	; GET ADDRESS OF NBUF
	0A	13	00EF	290	BEQL	30\$	; DEFAULTED (CONTINUOUS SAMPLING)
18	A6	61	D0	00F1	MOVL	(R1),IBF\$\$_NBUF(R6)	; STORE NBUF
	04	13	00F5	292	BEQL	30\$	; ZERO ALSO MEANS CONTINUOUS SAMPLING
4C	A6	04	A8	00F7	BISW	#FLG_M_CNTBFRS,IBF\$\$_FLAGS(R6)	; SET COUNT BUFFERS FLAG
			00FB	294			
			00FB	295			
		53	97	00FB	DECB	R3	; PROCESS MODE
		4C	19	00FD	BLSS	50\$	; DECR. ARG COUNT
		51	87	D0	MOVL	(R7)+,R1	; ARG. OMITTED
		47	13	0102	BEQL	50\$	; GET ADDRESS OF MODE
		51	61	3C	MOVZWL	(R1),R1	; DEFAULTED
				0107			; GET MODE
				0107			
04	51	05	E1	0107			; SET BITS IN MODE WORD (IN R5) DEPENDING ON BITS IN MODE ARG (IN R1)
00	55	0B	E2	010B	BBC	#5,R1,32\$	; BRANCH IF SERIAL
04	51	06	E1	010F	BBSS	#11,R5,32\$	; SET FOR PARALLEL
00	55	03	E2	0113	BBC	#6,R1,34\$	; BRANCH IF DEDICATED MODE
04	51	09	E1	0117	BBSS	#3,R5,34\$	; SET FOR MULTIREQUEST MODE
00	55	0A	E2	0118	BBC	#9,R1,36\$	; BRANCH IF CLOCK OVERFLOW TRIGGER
04	51	0A	E1	011F	BBSS	#10,R5,36\$	; SET FOR EXTERNAL TRIGGER
00	55	0F	E2	0123	BBC	#10,R1,38\$	; BRANCH IF NO TIME STAMPING
04	51	0B	E1	0127	BBSS	#15,R5,38\$	; SET FOR TIME STAMPING
00	55	0E	E2	012B	BBC	#11,R1,40\$	; BRANCH IF NO EVENT MARKING
04	51	0C	E1	012F	BBSS	#14,R5,40\$	; SET FOR EVENT MARKING
00	55	0C	E2	0133	BBC	#12,R1,42\$	; BRANCH IF IMMEDIATE START
04	51	0D	E1	0137	BBSS	#12,R5,42\$	; SET FOR DIGITAL INPUT START
00	55	05	E2	013B	BBC	#13,R1,44\$	; BRANCH IF SINGLE A/D CONVERTER
08	51	0E	E1	013F	BBSS	#5,R5,44\$	; SET FOR DUAL A/D CONVERTERS
4C	A6	10	A8	0143	BBC	#14,R1,50\$	; BRANCH IF OVER/UNDERRUN IS FATAL
00	55	17	E2	0147	BISW	#FLG_M_BFRORLSD,IBF\$\$_FLAGS(R6)	; SET BUFFER 0 RELEASED FLAG
				014B	BBSS	#23,R5,50\$	; SET FOR OVER/UNDERRUN NON-FATAL
				014B			
20	A6	55	C8	014B			; MODE WORD NOW COMPLETE EXCEPT FOR CHANNEL SELECTION BITS (BITS 8-9)
				014F	BISL	R5,IBF\$\$_CMDTBL+CMT\$\$_MODE(R6)	; OR INTO MODE WORD IN CMD TBL
				014F			
				014F			
		53	97	014F			; PROCESS DWELL
		09	19	0151	DECB	R3	; DECR. ARG COUNT
		51	87	D0	BLSS	60\$	; ARG OMITTED
		04	13	0156	MOVL	(R7)+,R1	; ADDRESS OF DWELL VALUE
3E	A6	61	B0	0158	BEQL	60\$	; DEFAULTED
				015C	MOVW	(R1),IBF\$\$_CMDTBL+CMT\$\$_DWELL(R6)	; STORE DWELL
				015C			
54	0000	8F	3C	015C			; PROCESS IEFN (EVENT FLAG OR COMPLETION ROUTINE ADDRESS)
		53	97	0161	MOVZWL	#IOS_STARTDATA!IOSM_SETEVF,R4	; I/O FUNCTION CODE
		17	19	0163	DECB	R3	; DECR. ARG COUNT
		50	87	D0	BLSS	65\$	; ARG OMITTED
		12	13	0165	MOVL	(R7)+,R0	; GET EVENT FLAG # OR ADDRESS OF ROUTINE
00000080	8F	50	D1	0168	BEQL	65\$	; EITHER DEFAULTED OR EVENT FLAG ZERO
		0C	1F	0171	CMPL	R0,#128	; EVENT FLAG OR AST ADDRESS?
		54	0000	8F	BLSSU	70\$	; EVENT FLAG
		10	A6	50	BICW	#IOSM_SETEVF,R4	; DON'T SET EVENT FLAG ON BUFFER FULLS
				0178	MOVL	R0,IBF\$\$_COMPLADDR(R6)	; SAVE COMPLETION ROUTINE ADDRESS
				017C			
		50	16	9A			; USE DEFAULT EVENT FLAG
				017C	MOVZBL	#DEFEVFLG,R0	
				017F			
				017F			; SAVE EVENT FLAG NUMBER IN R0

START SWEEP ROUTINES

4E A6	50	90	017F	346	MCVB	RO,IBF\$B_EFN(R6)	
			0183	347			
			0183	348			
	53	97	0183	349			
	09	19	0185	350			
	50	87	D0	0187	351		
		04	13	018A	352		
38 A6	60	B0	018C	353	MOVW	(R0),IBF\$L_CMDTBL+CMT\$W_DELAY(R6)	; STORE IN COMMAND TABLE
			0190	354			
			0190	355			
1A 4C A6	03	E0	0190	356	80\$:		
			0195	357	BBS	#FLG_V_SETADC,IBF\$W_FLAGS(R6),90\$	; BR. IF IT WAS CALLED
			0195	358			
	53	97	0195	359			
	09	19	0197	360			
	50	87	D0	0199	361		
		04	13	019C	362		
3A A6	60	90	019E	363	MOVW	(R0),IBF\$L_CMDTBL+CMT\$B_ICHN(R6)	; STORE ICHN
			01A2	364			
			01A2	365	85\$:		
	53	97	01A2	366			
	09	19	01A4	367			
	50	87	LJ	01A6	368		
		04	13	01A9	369		
3C A6	60	B0	01AB	370	MOVW	(R0),IBF\$L_CMDTBL+CMT\$W_NCHN(R6)	; STORE NCHN
			01AF	371			
			01AF	372	90\$:		
50	34	A6	D0	01AF	373		
		0F	13	0183	374		
			01B5	375			
			01B5	376			
	51	D4	01B5	377			
	51	02	C0	01B7	378	92\$:	
		80	B5	01BA	379		
		F9	18	01BC	380		
30 A6	51	D0	01BE	381	MOVW	R1,IBF\$L_CMDTBL+CMT\$L_RCLLEN(R6)	; STORE RCL LENGTH
		10	11	01C2	382		
			01C4	383			
3C A6	01	B1	01C4	384	94\$:		
	06	13	01C8	385			
21 A6	02	88	01CA	386			
	04	11	01CE	387			
21 A6	01	88	01D0	388	95\$:		
			01D4	389			
	53	D4	01D4	390	100\$:		
			01D6	391			
50	1C	A6	D0	01D6	392		
		08	13	01DA	393		
40 A6	60	7D	01DC	394			
53	06	A0	3C	01E0	395		
			01E4	396			
			01E4	397			
			01E4	398	110\$:		
52	4A	A6	3E	01E4	399		
		02F4	30	01E8	400		
	4B	50	E9	01EB	401		
			01EE	402			

START SWEEP ROUTINES

```

52 00000000'EF 9E 01EE 403 ; NOW SET UP TO DO QIO
53 00000000'EF 9E 01EE 404 MOVAB LPASS$CMPLTAST,R2 ; ADDRESS OF QIO COMPLETE AST
55 00000000'EF 9E 01F5 405 MOVAB LPASS$BFRAST,R3 ; ADDRESS OF BUFFER AST
01FC 406 MOVAB LPASS$OVRAST,R5 ; ADDRESS OF OVER/UNDERRUN AST
0203 407 $QIO_S IBF$B_EFN(R6),- ; EVENT FLAG
0203 408 IBF$W_CHAN(R6),- ; CHANNEL
0203 409 R4,- ; I/O FUNCTION CODE
0203 410 IBF$Q_IOSB(R6),- ; I/O STATUS BLOCK
0203 411 (R2),- ; COMPLETION AST ADDRESS
0203 412 R6,- ; AST PARAMETER (ADDRESS OF IBUF ARRAY)
0203 413 IBF$L_CMDTBL(R6),- ; ADDRESS OF COMMAND TABLE
0203 414 #40,- ; LENGTH OF COMMAND TABLE
0203 415 R3,- ; NORMAL BUFFER AST ADDRESS
0203 416 R5 ; OVER/UNDERRUN AST ADDRESS
10 50 E8 0226 417 BLBS R0,115$ ; SUCCESSFUL QIO
0229 418 ; ERROR IN QIO
50 DD 0229 419 PUSHL R0 ; SAVE STATUS
50 8ED0 022B 421 $DASSGN_S IBF$W_CHAN(R6) ; DEASSIGN CHANNEL
0236 422 POPL R0 ; RESTORE STATUS
0239 423 ; ALL ERRORS AND SUCCESS COME HERE WITH STATUS IN R0
0A 6C 91 0239 424 115$: CMPB (AP),#10 ; IND SPECIFIED?
51 28 AC 09 1F 023C 426 BLSSU 120$ ; NO
03 13 023E 427 MOVL 40(AP),R1 ; GET ADDRESS OF IND
61 50 D0 0242 428 BEQL 120$ ; DEFAULTED
0244 429 MOVL R0,(R1) ; STORE STATUS
0247 430
04 0247 431 120$: RET

```

LPA\$STPSWP - STOP SWEEP

```

0248 433 .SBTTL LPA$STPSWP - STOP SWEEP
0248 434 :++
0248 435 : FUNCTIONAL DESCRIPTION:
0248 436 :
0248 437 : THIS ROUTINE STOPS SWEEPS. A SWEEP CAN BE STOPPED IN TWO WAYS:
0248 438 : EITHER BY SETTING THE STOP BIT IN THE USER STATUS WORD (WHICH
0248 439 : STOPS AT THE END OF THE CURRENT BUFFER) OR BY ISSUING A CANCEL I/O
0248 440 : (WHICH STOPS IT IMMEDIATELY).
0248 441 :
0248 442 : CALLING SEQUENCE:
0248 443 :
0248 444 : CALLS/G
0248 445 :
0248 446 : INPUT PARAMETERS:
0248 447 :
0248 448 : IBUF(AP) ADDRESS OF IBUF ARRAY
0248 449 : IWHEN(AP) ADDRESS OF BYTE WHICH SPECIFIES WHEN TO
0248 450 : STOP SWEEP (0 = IMMED. NON-0 = AT END OF BFR.)
0248 451 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE STATUS
0248 452 :
0248 453 : IMPLICIT INPUTS:
0248 454 :
0248 455 : VARIOUS FIELDS IN IBUF ARRAY
0248 456 :
0248 457 : OUTPUT PARAMETERS:
0248 458 :
0248 459 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE STATUS
0248 460 :
0248 461 : IMPLICIT OUTPUTS:
0248 462 :
0248 463 : NONE
0248 464 :
0248 465 : COMPLETION CODES:
0248 466 :
0248 467 : 1 INDICATES SUCCESS
0248 468 : VARIOUS ERRORS RETURNED BY $CANCEL
0248 469 :
0248 470 : SIDE EFFECTS:
0248 471 :
0248 472 : FIELDS IN THE IBUF ARRAY MAY BE MODIFIED
0248 473 :
0248 474 :--
0248 475 :
0040 0248 476 .ENTRY LPA$STPSWP,^M<R6>
024A 477
56 04 AC D0 024A 478 MOVL 4(AP),R6 ; GET ADDRESS OF IBUF ARRAY
024E 479
02 6C 91 024E 480 CMPB (AP),#2 ; IS IWHEN SUPPLIED?
14 1F 0251 481 BLSSU 20$ ; NO
50 08 AC D0 0253 482 MOVL 8(AP),R0 ; GET ADDRESS OF IWHEN
0E 13 0257 483 BEQL 20$ ; DEFAULTED
60 D5 0259 484 TSTL (R0) ; DETERMINE WHEN
0A 13 025B 485 BEQL 20$ ; IMMEDIATELY
025D 486
; AT END OF CURRENT BUFFER - SET STOP BIT IN USW
49 A6 40 8F 88 025D 487 BISB #^X40,IBF$_USW+1(R6)
50 01 D0 025D 488 MOVL #1,R0 ; SUCCESS
0262 489

```

LPA\$STPSWP - STOP SWEEP

0B	11	0265	490	6RB	40\$		
		0267	491				
		0267	492	20\$:		: ISSUE CANCEL I/O	
		0267	493			\$CANCEL_S	IBFSW_CHAN(R6) ; RETURNS STATUS IN R0
		0272	494				
03	6C	91	0272	495	40\$:	CMPB	(AP),#3 ; IND SUPPLIED?
	09	1F	0275	496		BLSSU	60\$ ; NO
51	0C	AC	D0	0277	497	MOVL	12(AP),R1 ; YES, GET ADDRESS
	03	13	027B	498		BEQL	60\$ ; DEFAULTED
61	50	D0	027D	499		MOVL	RO,(R1) ; STORE STATUS IN IND
	04	0280	500	60\$:		RET	

```

0281 502 .SBTTL LPAS$CLOCKA - SET CLOCK A RATE
0281 503 :++
0281 504 : FUNCTIONAL DESCRIPTION:
0281 505 :
0281 506 : THIS ROUTINE SETS THE RATE FOR CLOCK A
0281 507 :
0281 508 : CALLING SEQUENCE:
0281 509 :
0281 510 : CALLS/G
0281 511 :
0281 512 : INPUT PARAMETERS:
0281 513 :
0281 514 : IRATE(AP) ADDRESS OF LONGWORD CONTAINING CLOCK RATE
0281 515 : IPRSET(AP) ADDRESS OF WORD CONTAINING CLOCK PRESET
0281 516 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
0281 517 : CHAN(AP) ADDRESS OF WORD CONTAINING NUMBER TO IDENTIFY
0281 518 : WHICH LPA-11
0281 519 :
0281 520 : IMPLICIT INPUTS:
0281 521 :
0281 522 : NONE
0281 523 :
0281 524 : OUTPUT PARAMETERS:
0281 525 :
0281 526 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
0281 527 :
0281 528 : IMPLICIT OUTPUTS:
0281 529 :
0281 530 : NONE
0281 531 :
0281 532 : COMPLETION CODES:
0281 533 :
0281 534 : 1 INDICATES SUCCESS
0281 535 : VARIOUS VMS SYSTEM STATUS CODES INDICATE ERRORS
0281 536 :
0281 537 : SIDE EFFECTS:
0281 538 :
0281 539 : NONE
0281 540 :
0281 541 :--
0281 542 :
00FC 0281 543 .ENTRY LPAS$CLOCKA,^M<R2,R3,R4,R5,R6,R7>
0283 544
55 01 D0 0283 545 MOVL #1,R5 ; MODE WORD
0286 546
0286 547 : BUILD CLOCK STATUS. IF IRATE IS >= 0 THEN USE AS IS. IF
0286 548 : IRATE IS < 0 THEN SET SCHMITT TRIGGER 1 INTERRUPT ENABLE (BIT 14).
56 00004141 8F D0 0286 549 MOVL #^X4141,R6 ; CLOCK STATUS INCLUDING BIT 14
50 04 BC D0 028D 550 MOVL @4(AP),R0 ; PUT IRATE IN R0
56 4000 8F AA 0291 551 BLSS 10$ ; ITS < 0. LEAVE RATE = 0 (BITS 1 - 3)
56 03 01 50 F0 0293 552 BICW #^X4000,R6 ; ITS >= 0. CLEAR BIT 14
57 08 BC 3C 0298 553 INSV R0,#1,#3,R6 ; INSERT RATE
10$: 029D 554 MOVZWL @8(AP),R7 ; CLOCK PRESET
02A1 555
02A1 556 : PUT # OF ARGS LEFT IN R0, POINTER TO THEM IN R1, AND
02A1 557 : JOIN COMMON CLOCK ROUTINE
50 6C 02 83 02A1 558 SUBB3 #2,(AP),R0 ; # OF ARGS LEFT

```

LPASSWEEP  
V04-000

LPASCLOCKA - SET CLOCK A RATE

C 11

16-SEP-1984 01:44:18 VAX/VMS Macro V04-00  
5-SEP-1984 01:32:23 [IOSUP.SRC]LASWEEP.MAR;1

Page 13  
(6)

LPA  
V04.

51	OC	AC	DE	02A5	559	MOVAL	12(AP),R1
		38	11	02A9	560	BRB	CLKCOM

: ADDRESS OF NEXT ARG  
: JOIN COMMON ROUTINE



LPA\$CLOCKB - SET CLOCK B RATE

```

02AB 562 .SBTTL LPA$CLOCKB - SET CLOCK B RATE
02AB 563 :++
02AB 564 : FUNCTIONAL DESCRIPTION:
02AB 565 :
02AB 566 : THIS ROUTINE SETS THE RATE FOR CLOCK B
02AB 567 :
02AB 568 : CALLING SEQUENCE:
02AB 569 :
02AB 570 : CALLS/G
02AB 571 :
02AB 572 : INPUT PARAMETERS:
02AB 573 :
02AB 574 : IRATE(AP) ADDRESS OF LONGWORD CONTAINING CLOCK RATE
02AB 575 : IPRSET(AP) ADDRESS OF WORD CONTAINING CLOCK PRESET
02AB 576 : MODE(AP) ADDRESS OF A WORD WHICH SPECIFIES OPTIONS
02AB 577 : BIT 0 SET INDICATES OPERATE CLOCK B IN
02AB 578 : NON-INTERRUPT MODE
02AB 579 : BIT 1 SET INDICATES THE FEED B TO A BIT SHOULD
02AB 580 : BE SET IN THE CLOCK B STATUS REGISTER
02AB 581 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
02AB 582 : CHAN(AP) ADDRESS OF WORD CONTAINING NUMBER TO IDENTIFY
02AB 583 : WHICH LPA-11
02AB 584 :
02AB 585 : IMPLICIT INPUTS:
02AB 586 :
02AB 587 : NONE
02AB 588 :
02AB 589 : OUTPUT PARAMETERS:
02AB 590 :
02AB 591 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION STATUS
02AB 592 :
02AB 593 : IMPLICIT OUTPUTS:
02AB 594 :
02AB 595 : NONE
02AB 596 :
02AB 597 : COMPLETION CODES:
02AB 598 :
02AB 599 : 1 INDICATES SUCCESS
02AB 600 : VARIOUS VMS SYSTEM STATUS CODES INDICATE ERRORS
02AB 601 :
02AB 602 : SIDE EFFECTS:
02AB 603 :
02AB 604 : NONE
02AB 605 :
02AB 606 : --
02AB 607 :
00FC 02AB 608 .ENTRY LPA$CLOCKB,^M<R2,R3,R4,R5,R6,R7>
02AD 609
56 55 11 D0 02AD 610 MOVL #^X11,R5 ; MODE WORD
00000041 8F D0 02BC 611 MOVL #^X41,R6 ; CLOCK STATUS
50 04 AC D0 02B7 612 MOVL 4(AP),R0 ; ADDRESS OF IRATE
1E 13 02BB 613 BEQL 40$ ; DEFAULTED
50 60 D0 02BD 614 MOVL (R0),R0 ; GET IRATE
19 13 02C0 615 BEQL 40$ ; ZERO
56 03 01 50 F0 02C2 616 INSV R0,#1,#3,R6 ; INSERT IRATE
04 0C BC E9 02C7 617 BLBC @12(AP),20$ ; BR. IF LEAVE INTERRUPTS ENABLED
56 40 8F 8A 02CB 618 BICB #^X40,R6 ; CLEAR INTERRUPT ENABLE

```



LPA\$LAMSKS - SET MASKS BUFFER

```

0351 673 .SBTTL LPA$LAMSKS - SET MASKS BUFFER
0351 674 :++
0351 675 : FUNCTIONAL DESCRIPTION:
0351 676 :
0351 677 : THIS ROUTINE PERFORMS TWO COMPLETELY SEPARATE FUNCTIONS.
0351 678 : FIRST IT MUST BE CALLED BY PROGRAMS THAT UTILIZE TWO OR MORE
0351 679 : LPA-11S TO IDENTIFY WHICH LPA-11 IS TO BE USED IN A SUBSEQUENT
0351 680 : START SWEEP CALL. THE SECOND ARGUMENT IS A NUMBER WHICH IS APPENDED
0351 681 : TO THE LOGICAL NAME THAT IS USED TO ASSIGN A CHANNEL TO. IT IS
0351 682 : ASSUMED THAT THE USER HAS ASSIGNED THE RESULTANT LOGICAL NAME
0351 683 : TO THE APPROPRIATE LPA-11.
0351 684 : SECONDLY, THIS ROUTINE MUST BE CALLED BY PROGRAMS THAT UTILIZE
0351 685 : DIGITAL INPUT STARTING OR EVENT MARKING. ARGUMENTS CAN SUPPLIED
0351 686 : FOR THE DIGITAL INPUT START WORD AND MASK AND FOR THE EVENT MARK
0351 687 : WORD AND MASK.
0351 688 :
0351 689 : CALLING SEQUENCE:
0351 690 :
0351 691 : CALLS/G
0351 692 :
0351 693 : INPUT PARAMETERS:
0351 694 :
0351 695 : LAMSKB(AP) ADDRESS OF 8 BYTE ARRAY
0351 696 : NUM(AP) ADDRESS OF WORD CONTAINING NUMBER TO APPEND TO LOG. NAME
0351 697 : IUNIT(AP) UNUSED (PRESENT FOR RSX-11M COMPATIBILITY)
0351 698 : STWRDN(AP) ADDRESS OF BYTE CONTAINING DIGITAL START CHANNEL
0351 699 : EVMRKN(AP) ADDRESS OF BYTE CONTAINING EVENT MARK CHANNEL
0351 700 : STWRDM(AP) ADDRESS OF WORD CONTAINING DIGITAL START MASK
0351 701 : EVMRKM(AP) ADDRESS OF WORD CONTAINING EVENT MARK MASK
0351 702 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE STATUS
0351 703 :
0351 704 : IMPLICIT INPUTS:
0351 705 :
0351 706 : NONE
0351 707 :
0351 708 : OUTPUT PARAMETERS:
0351 709 :
0351 710 : IND(AP) ADDRESS OF LONGWORD TO RECEIVE STATUS
0351 711 :
0351 712 : IMPLICIT OUTPUTS:
0351 713 :
0351 714 : THE 8 BYTE ARRAY IS FILLED IN
0351 715 :
0351 716 : COMPLETION CODES:
0351 717 :
0351 718 : 1 INDICATES SUCCESS (THIS ROUTINE ALWAYS RETURNS SUCCESS.
0351 719 : THE ARGUMENT IS PRESENT FOR COMPATIBILITY ONLY)
0351 720 :
0351 721 : SIDE EFFECTS:
0351 722 :
0351 723 : NONE
0351 724 :
0351 725 :--
0351 726 :
000C 0351 727 .ENTRY LPA$LAMSKS,^M<R2,R3>
0353 728
53 04 AC DE 0353 729 MOVAL 4(AP),R3 ; R3 CONTAINS ADDRESS OF ARGUMENT LIST

```

PSE

\$AB  
\_LP

Pha

---  
Ini  
Com  
Pas  
Sym  
Pas  
Sym  
Pse  
Cro  
Ass

The  
280  
The  
135  
20

Mac

---  
\_S2

195

The  
MAC

LPASLAMS - SET MASKS BUFFER

```

52 83 D0 0357 730      MOVL  (R3)+,R2      ; R2 CONTAINS ADDRESS OF 8 BYTE ARRAY
    62 7C 035A 731      CLRQ  (R2)          ; CLEAR ARRAY
    035C 732
    035C 733      ; PROCESS NUM
51 6C 02 83 035C 734      SUBB3  #2,(AP),R1    ; R1 CONTAINS # OF REMAINING ARGS
    4B 19 0360 735      BLSS  90$                ; NO MORE ARGS
    50 83 D0 0362 736      MOVL  (R3)+,R0          ; GET ADDRESS OF NUM
    04 13 0365 737      BEQL  10$                ; DEFAULTTED
06 A2 60 B0 0367 738      MOVW  (R0),6(R2)        ; STORE IN ARRAY
    036B 739
    036B 740 10$:      ; PROCESS STWRDN
    51 02 82 036B 741      SUBB  #2,R1          ; ENOUGH ARGS?
    3D 19 036E 742      BLSS  90$                ; NOPE
    83 D5 0370 743      TSTL  (R3)+          ; SKIP OVER UNUSED ARG
    50 83 D0 0372 744      MOVL  (R3)+,R0          ; GET ADDRESS OF STWRDN
    03 13 0375 745      BEQL  20$                ; DEFAULTTED
    62 60 90 0377 746      MOVB  (R0),(R2)        ; STORE IN ARRAY
    037A 747
    037A 748 20$:      ; PROCESS EVMRKN
    51 97 037A 749      DECB  R1          ; ENOUGH ARGS?
    2F 19 037C 750      BLSS  90$                ; NOPE
    50 83 D0 037E 751      MOVL  (R3)+,R0          ; GET ADDRESS OF EVMRKN
    04 13 0381 752      BEQL  30$                ; DEFAULTTED
01 A2 60 90 0383 753      MOVB  (R0),1(R2)        ; STORE IN ARRAY
    0387 754
    0387 755 30$:      ; PROCESS STWRDM
    51 97 0387 756      DECB  R1          ; ENOUGH ARGS?
    22 19 0389 757      BLSS  90$                ; NOPE
    50 83 D0 038B 758      MOVL  (R3)+,R0          ; GET ADDRESS OF STWRDM
    04 13 038E 759      BEQL  40$                ; DEFAULTTED
02 A2 60 B0 0390 760      MOVW  (R0),2(R2)        ; STORE IN ARRAY
    0394 761
    0394 762 40$:      ; PROCESS EVMRKM
    51 97 0394 763      DECB  R1          ; ENOUGH ARGS?
    15 19 0396 764      BLSS  90$                ; NOPE
    50 83 D0 0398 765      MOVL  (R3)+,R0          ; GET ADDRESS OF EVMRKN
    04 13 039B 766      BEQL  50$                ; DEFAULTTED
04 A2 60 B0 039D 767      MOVW  (R0),4(R2)        ; STORE IN ARRAY
    03A1 768
    03A1 769 50$:      ; PROCESS IND
    51 97 03A1 770      DECB  R1          ; ENOUGH ARGS?
    08 19 03A3 771      BLSS  90$                ; NOPE
    50 83 D0 03A5 772      MOVL  (R3)+,R0          ; GET ADDRESS OF IND
    03 13 03A8 773      BEQL  90$                ; DEFAULTTED
    60 01 D0 03AA 774      MOVL  #1,(R0)        ; STORE SUCCESS STATUS
    03AD 775
    04 03AD 776 90$:  RET

```

LPASSETADC - SET CHANNEL PARAMETERS

```

03AE 778      .SBTTL LPASSETADC - SET CHANNEL PARAMETERS
03AE 779      :++
03AE 780      : FUNCTIONAL DESCRIPTION:
03AE 781      :
03AE 782      :     THIS ROUTINE SETS THE CHANNEL SAMPLING PARAMETERS.
03AE 783      :
03AE 784      : CALLING SEQUENCE:
03AE 785      :
03AE 786      :     CALLS/G
03AE 787      :
03AE 788      : INPUT PARAMETERS:
03AE 789      :
03AE 790      :     IBUF(AP)      ADDRESS OF IBUF ARRAY
03AE 791      :     IFLAG(AP)   UNUSED (PRESENT FOR COMPATIBILITY WITH RSX-11M)
03AE 792      :     ICHN(AP)    IF INC IS DEFAULTED OR NON-ZERO, THIS IS THE ADDRESS
03AE 793      :                OF A BYTE CONTAINING THE INITIAL CHANNEL NUMBER.
03AE 794      :                IF INC = 0, THIS IS THE ADDRESS OF A RANDOM
03AE 795      :                CHANNEL LIST.
03AE 796      :     NCHN(AP)     ADDRESS OF A WORD CONTAINING NUMBER OF SAMPLES TO
03AE 797      :                BE TAKEN PER SAMPLE SEQUENCE.
03AE 798      :     INC(AP)      ADDRESS OF A BYTE CONTAINING THE CHANNEL INCREMENT.
03AE 799      :                IF THIS BYTE CONTAINS 0, THEN ICHN IS THE ADDRESS
03AE 800      :                OF A RANDOM CHANNEL LIST.
03AE 801      :     IND(AP)      ADDRESS OF A LONGWORD TO RECEIVE STATUS.
03AE 802      :
03AE 803      : IMPLICIT INPUTS:
03AE 804      :
03AE 805      :     NONE
03AE 806      :
03AE 807      : OUTPUT PARAMETERS:
03AE 808      :
03AE 809      :     IND(AP)      ADDRESS OF A LONGWORD TO RECEIVE STATUS
03AE 810      :
03AE 811      : IMPLICIT OUTPUTS:
03AE 812      :
03AE 813      :     NONE
03AE 814      :
03AE 815      : COMPLETION CODES:
03AE 816      :
03AE 817      :     0            INDICATES LPASSETIBF WAS NOT CALLED PRIOR TO THIS CALL
03AE 818      :     1            INDICATES SUCCESS
03AE 819      :
03AE 820      : SIDE EFFECTS:
03AE 821      :
03AE 822      :     VARIOUS FIELDS IN THE IBUF ARRAY ARE MODIFIED
03AE 823      :
03AE 824      :--
03AE 825      :
005C 03AE 826      .ENTRY LPASSETADC,^M<R2,R3,R4,R6>
03AE 827      :
03AE 828      : CLRL    R0          : STATUS
03AE 829      : MOVL   4(AP),R6    : ADDRESS OF IBUF ARRAY
03AE 830      : CMPW  IBF$Q_I0ST(R6),#INITCODE : VERIFY LPASSETIBF WAS CALLED
03AE 831      : BNEQ  80$          : IT WASN'T - ERROR
03AE 832      : MOVAL  12(AP),R3   : POINT TO ICHN ARG
03AE 833      :
03AE 834      : PROCESS ICHN

```

```

56 04 AC D4
1234 8f 66 B1
3D 12
53 0C AC DE

```

LPASSETADC - SET CHANNEL PARAMETERS

```

52 6C 03 83 03C1 835 SUBB3 #3 (AP),R2 ; ENOUGH ARGS?
      2D 19 03C5 836 BLSS 70$ ; NOPE
54 83 D0 03C7 837 MOVL (R3)+,R4 ; R4 = ADDRESS OF ICHN OR RCL
      03CA 838 ; PROCESS NCHN
      03CA 839
      52 97 03CA 840 DECB R2 ; ENOUGH ARGS?
      1E 19 03CC 841 BLSS 20$ ; NOPE
51 83 D0 03CE 842 MOVL (R3)+,R1 ; GET ADDRESS OF NCHN
      04 13 03D1 843 BEQL 10$ ; DEFAULTED
3C A6 61 B0 03D3 844 MOVW (R1),IBF$$_CMDTBL+CMT$$_NCHN(R6) ; STORE NCHN
      03D7 845
      03D7 846 10$: ; PROCESS INC
      52 97 03D7 847 DECB R2 ; ENOUGH ARGS?
      11 19 03D9 848 BLSS 20$ ; NOPE
51 83 D0 03DB 849 MOVL (R3)+,R1 ; GET ADDRESS OF INC
      0C 13 03DE 850 BEQL 20$ ; DEFAULTED
3B A6 61 90 03E0 851 MOVW (R1),IBF$$_CMDTBL+CMT$$_INC(R6) ; STORE INC
      06 12 03E4 852 BNEQ 20$ ; NON-ZERO, SO ICHN IS NOT RCL ADDR.
      03E6 853
      03E6 854 ; INC = 0 SO ICHN IS RCL ADDRESS (IN R4)
34 A6 54 D0 03E6 855 MOVL R4,IBF$$_CMDTBL+CMT$$_RCLADDR(R6) ; STORE RCL ADDRESS
      08 11 03EA 856 BRB 70$
      03EC 857
      03EC 858 20$: ; INC WAS EITHER DEFAULTED OR NON-ZERO, SO R4 POINTS TO ICHN VALUE
      54 D5 03EC 859 TSTL R4 ; WAS ICHN DEFAULTED?
      04 13 03EE 860 BEQL 70$ ; YES
3A A6 64 90 03F0 861 MOVW (R4),IBF$$_CMDTBL+CMT$$_ICHN(R6) ; NO, STORE ICHN VALUE
      03F4 862
      03F4 863 70$: ; SUCCESS RETURN
4C A6 08 A8 03F4 864 BISW #FLG_M_SETADC,IBF$$_FLAGS(R6) ; SET SETADC CALLED BIT
      50 D6 03F8 865 INCL R0 ; SET SUCCESS CODE
      03FA 866
      03FA 867 80$: ; STORE COMPLETION CODE IN IND
      06 6C 91 03FA 868 CMPB (AP),#6 ; ENOUGH ARGS?
      09 1F 03FD 869 BLSSU 90$ ; NO
51 18 AC D0 03FF 870 MOVL 24(AP),R1 ; GET ADDRESS OF IND
      03 13 0403 871 BEQL 90$ ; DEFAULTED
      61 50 D0 0405 872 MOVL R0,(R1) ; STORE VALUE IN IND
      04 0408 873 90$: RET

```

LPASCVADF - CONVERT A/D TO FLOATING POIN

```

0409 875 .SBTTL LPASCVADF - CONVERT A/D TO FLOATING POINT
0409 876 .SBTTL LPASFLT16 - CONVERT UNSIGNED WORD TO FLOATING POINT
0409 877 :++
0409 878 : FUNCTIONAL DESCRIPTION:
0409 879 :
0409 880 : LPASCVADF CONVERTS A NUMBER RETURNED BY AN A/D CONVERTER TO
0409 881 : FLOATING POINT. THE NUMBER IS A SIGNED 12 BIT NUMBER WITH BIT 11
0409 882 : THE SIGN BIT. IT IS FOR THIS REASON THAT ^X0800 IS SUBTRACTED
0409 883 : FROM THE NUMBER BEFORE IT IS CONVERTED TO FLOATING POINT. NOTE THAT
0409 884 : THE ORIGINAL CVADF ROUTINE (ON RSX-11M?) EXPECTED AN A/D VALUE
0409 885 : IN BITS 0 - 11 OF THE WORD AND A GAIN IN BITS 12 - 15. HOWEVER,
0409 886 : THE LPA-11 ON VAX DOES NOT SUPPORT ANY A/D'S THAT SUPPLY A GAIN
0409 887 : IN BITS 12 - 15. THEREFORE, THIS ROUTINE DOES NOT USE THOSE BITS
0409 888 : AS A GAIN.
0409 889 : LPASFLT16 CONVERTS AN UNSIGNED(!) WORD TO FLOATING POINT AND IS
0409 890 : INCLUDED HERE FOR COMPATIBILITY REASONS.
0409 891 :
0409 892 : CALLING SEQUENCE:
0409 893 :
0409 894 : CALLS/CALLG
0409 895 : THESE ROUTINES MAY BE CALLED AS FUNCTIONS
0409 896 :
0409 897 : INPUT PARAMETERS:
0409 898 :
0409 899 : IVAL(AP) ADDRESS OF WORD TO CONVERT
0409 900 : VAL(AP) ADDRESS OF LONGWORD TO RECEIVE RESULT
0409 901 :
0409 902 : IMPLICIT INPUTS:
0409 903 :
0409 904 : NONE
0409 905 :
0409 906 : OUTPUT PARAMETERS:
0409 907 :
0409 908 : VAL(AP) ADDRESS OF LONGWORD TO RECEIVE RESULT
0409 909 :
0409 910 : IMPLICIT OUTPUTS:
0409 911 :
0409 912 : NONE
0409 913 :
0409 914 : COMPLETION CODES:
0409 915 :
0409 916 : NONE
0409 917 :
0409 918 : SIDE EFFECTS:
0409 919 :
0409 920 : NONE
0409 921 :--
0409 922 :
0409 923 :
0409 924 .ENTRY LPASFLT16,^M<>
50 04 BC 0000 040B 925 MOVZWL @4(AP),RO ; CONVERT INPUT WORD TO LONGWORD
50 50 50 4E 040F 926 CVTLF RO,RO ; CONVERT TO FLOATING POINT
0409 927 BRB CVCOM
0414 928
0414 929 .ENTRY LPASCVADF,^M<>
50 04 BC 0800 8F A3 0416 930 SUBW3 #^X0800,@4(AP),RO ; SUBTRACT ^X0800 FROM INPUT
50 50 50 4D 041D 931 CVTWF RO,RO ; CONVERT TO FLOATING

```

			0420	932			
			0420	933			
			0420	934	CVCOM:	: OPTIONALLY STORE RESULT	
02	6C	91	0420	935		: (MPB (AP),#2	: ENOUGH ARGS SUPPLIED?
	09	1F	0423	936		: BLSSU 90\$	: NO
51	08	AC	D0 0425	937		: MOVL 8(AP),R1	: GET ADDRESS OF ARG
	03	13	0429	938		: BEQL 90\$	: DEFAULTED
61	50	D0	042B	939		: MOVL R0,(R1)	: STORE RESULT
		04	042E	940	90\$:	: RET	

Pse

---

COM

DAT

COD

\_LI



LPASXRATE - COMPUTE CLOCK RATE AND PRESE

```
042F 942 .SBTTL LPASXRATE - COMPUTE CLOCK RATE AND PRESET
042F 943 :++
042F 944 : FUNCTIONAL DESCRIPTION:
042F 945 :
042F 946 : THIS ROUTINE COMPUTES A CLOCK RATE AND PRESET GIVEN A DESIRED
042F 947 : DWELL (INTER-SAMPLE INTERVAL). THE CLOCK RATE IS ALWAYS
042F 948 : THE HIGHEST RATE WHICH WILL PERMIT THE REQUESTED DWELL IN ORDER TO
042F 949 : ACHIEVE THE FINEST RESOLUTION. THIS MAY BE DIFFERENT FOR CLOCK A
042F 950 : AND CLOCK B AS THEY HAVE DIFFERENT MAXIMUM PRESETS. THE CLOCK RATE
042F 951 : IS RETURNED AS A NUMBER (1 - 5) WHICH CAN THEN BE USED AS THE CLOCK
042F 952 : RATE FOR LPASCLOCKA OR LPASCLOCKB. IF CALLED AS A FUNCTION, THE
042F 953 : FUNCTION VALUE IS THE ACTUAL DWELL BEING SUPPLIED, WHICH MAY DIFFER
042F 954 : FROM THE REQUESTED DWELL DUE TO TRUNCATION ERROR.
042F 955 :
042F 956 : CALLING SEQUENCE:
042F 957 :
042F 958 : CALLS/G
042F 959 : MAY BE CALLED AS A FUNCTION
042F 960 :
042F 961 : INPUT PARAMETERS:
042F 962 :
042F 963 : DWELL(AP) ADDRESS OF LONGWORD CONTAINING DWELL AS
042F 964 : A FLOATING POINT NUMBER
042F 965 : IRATE(AP) ADDRESS OF A LONGWORD TO RECEIVE THE CLOCK
042F 966 : RATE (1 - 5) (0 INDICATES ERROR)
042F 967 :
042F 968 : IPRSET(AP) ADDRESS OF A WORD TO RECEIVE CLOCK PRESET
042F 969 : IFLAG(AP) ADDRESS OF A BYTE WHICH INDICATES WHETHER
042F 970 : THE COMPUTATION IS FOR CLOCK A (MAXIMUM
042F 971 : PRESET = 65535) OR CLOCK B (MAXIMUM PRESET =
042F 972 : 255) 0 = CLOCK A. NON-0 = CLOCK B.
042F 973 : IMPLICIT INPUTS:
042F 974 :
042F 975 : NONE
042F 976 :
042F 977 : OUTPUT PARAMETERS:
042F 978 :
042F 979 : IRATE(AP) SEE ABOVE
042F 980 : IPRSET(AP) SEE ABOVE
042F 981 : R0 FUNCTION VALUE. ACTUAL DWELL COMPUTED
042F 982 : AS A FLOATING POINT NUMBER.
042F 983 :
042F 984 : NOTE THAT IF THE DESIRED DWELL IS TOO SMALL OR TOO LARGE TO BE
042F 985 : ACHIEVED, THEN BOTH IRATE(AP) AND R0 WILL CONTAIN ZERO
042F 986 :
042F 987 : IMPLICIT OUTPUTS:
042F 988 :
042F 989 : NONE
042F 990 :
042F 991 : COMPLETION CODES:
042F 992 :
042F 993 : IF IRATE(AP) OR R0 EQUALS ZERO, THEN THE DESIRED DWELL COULD
042F 994 : NOT BE ACHIEVED.
042F 995 :
042F 996 : SIDE EFFECTS:
042F 997 :
042F 998 : NONE
```

LPASXRATE - COMPUTE CLOCK RATE AND PRESE

```

042F 999 :
042F 1000 :--
042F 1001 :
003C 042F 1002 .ENTRY LPASXRATE,^M<R2,R3,R4,R5>
0431 1003 ; GET MAXIMUM PRESET
0431 1004 ; MOVF #^F65535.0,R5 ; MAXIMUM PRESET FOR CLOCK A
55 FF00487F 8F 50 0431 1005 ; TSTB @16(AP) ; TEST FLAG
10 BC 95 0438 1006 ; BEQL 10$ ; COMPUTATION IS FOR CLOCK A
07 13 043B 1007 ; MOVF #^F255.0,R5 ; MAXIMUM PRESET FOR CLOCK B
55 0000447F 8F 50 043D 1008
0444 1009
54 04 04 50 0444 1010 10$: MOVF @4(AP),R4 ; DESIRED DWELL
37BD3686 8F 54 51 0448 1011 ; CMPF R4,#^F0.000001 ; IS IT WITHIN RANGE?
07 13 044F 1012 ; BLSS 40$ ; NO
53 24004A74 8F 50 0451 1013 ; MOVF #^F1000000.0,R3 ; MAXIMUM CLOCK RATE (1 MHZ)
52 01 D0 0458 1014 ; MOVL #1,R2 ; LOOP COUNTER AND CLOCK RATE NUMBER
045B 1015
045B 1016 20$: ; CALCULATE PRESET: PRESET = RATE X DWELL
51 53 54 45 045B 1017 ; MULF3 R4,R3,R1 ; CALCULATED PRESET IN R1
55 51 51 045F 1018 ; CMPF R1,R5 ; LESS THAN MAXIMUM PRESET ALLOWED?
09 15 0462 1019 ; BLEQ 30$ ; YES - USE IT
53 22 46 0464 1020 ; DIVF #^F10.0,R3 ; NO - DIVIDE CLOCK RATE BY 10
FO 52 05 F3 0467 1021 ; AOBLEQ #5,R2,20$ ; AND TRY NEXT CLOCK RATE
046B 1022
046B 1023 ; IF WE FALL THROUGH THAN DESIRED DWELL IS TOO GREAT TO BE
046B 1024 ; ACHIEVED WITH THE SLOWEST CLOCK RATE AND LARGEST PRESET.
16 11 046B 1025 ; BRB 40$
046D 1026
046D 1027 30$: ; HAVE CALCULATED PRESET IN R1. ADD 0.5 AND TRUNCATE TO AN INTEGER
51 00 40 046D 1028 ; ADDF #^F0.5,R1 ; TO ROUND
51 51 4A 0470 1029 ; CVTFL R1,R1 ; CVT TO A LONGWORD TO AVOID OVERFLOW
51 51 3C 0473 1030 ; MOVZWL R1,R1 ; NOW TRUNCATE TO A WORD
0476 1031
0476 1032 ; STORE PRESET (AS TWO'S COMPLEMENT OF CALC. PRESET BECAUSE THAT'S
0476 1033 ; WHAT THE CLOCKS USE)
OC BC 51 AE 0476 1034 ; MNEGW R1,@12(AP)
047A 1035
047A 1036 ; NOW CONVERT EVERYTHING BACK TO FLOATING POINT TO GIVE CALLER
047A 1037 ; ACTUAL DWELL COMPUTED (WHICH MAY BE DIFFERENT DUE TO TRUNCATION ERROR)
51 51 51 4E 047A 1038 ; CVTLF R1,R1 ; CONVERT TO FLOATING POINT
50 51 53 47 047D 1039 ; DIVF3 R3,R1,R0 ; ACTUAL DWELL = PRESET / RATE
04 11 0481 1040 ; BRB 50$
0483 1041
0483 1042 40$: ; ERROR - EITHER DESIRED DWELL WAS TOO GREAT (> 655.35 FOR CLOCK A
52 D4 0483 1044 ; OR > 2.55 FOR CLOCK B) OR WAS TOO SMALL (< 0.000001 FOR EITHER CLOCK)
50 D4 0485 1045 ; CLRL R2 ; RETURN 0 FOR CLOCK RATE NUMBER
0487 1046 ; CLRL R0 ; AND 0 FOR COMPUTED DWELL
08 BC 52 D0 0487 1047 50$: ; STORE CLOCK RATE NUMBER (IN R2)
04 04 0487 1048 ; MOVL R2,@8(AP)
0488 1049 ; RET ; COMPUTED DWELL IN R0 FOR FUNCTION CALL

```

Sym  
---  
ABO  
ABO  
AFT  
ALL  
ALL  
ALL  
APP  
APP  
APP  
APP  
BAT  
BRO  
BUG  
CHE  
CHE  
CHE  
CHE  
CHK  
CLO  
COM  
COM  
CRE  
CRE  
CRE  
CRE  
CRE  
CRE  
CRE  
CTL  
CTL  
DEA  
DEA  
DEA  
DEA  
DEL  
DEL  
DEL  
DEL

LPASLOADMC - LOAD MICROCODE

```

048C 1051 .SBTTL LPASLOADMC - LOAD MICROCODE
048C 1052 :++
048C 1053 : FUNCTIONAL DESCRIPTION:
048C 1054 :
048C 1055 : THIS ROUTINE SENDS A REQUEST TO THE LPA-11 MICROCODE LOADER
048C 1056 : PROCESS TO LOAD A SPECIFIED VERSION OF MICROCODE INTO A SPECIFIED
048C 1057 : LPA-11. THE LPA-11 IS SPECIFIED BY A NUMBER WHICH IS APPENDED
048C 1058 : TO A LOGICAL NAME (SEE LPASS$ASSIGN ROUTINE)
048C 1059 :
048C 1060 : CALLING SEQUENCE:
048C 1061 :
048C 1062 : CALLS/G
048C 1063 :
048C 1064 : INPUT PARAMETERS:
048C 1065 :
048C 1066 : ITYPE(AP) ADDRESS OF BYTE CONTAINING TYPE OF
048C 1067 : MICROCODE TO LOAD
048C 1068 : 1 = MULTIREQUEST
048C 1069 : 2 = DED. A/D
048C 1070 : 3 = DED. D/A
048C 1071 : INUM(AP) ADDRESS OF WORD CONTAINING NUMBER TO APPEND
048C 1072 : TO LOGICAL NAME (TO ASSIGN CHANNEL TO)
048C 1073 : ISTAT(AP) ADDRESS OF LONGWORD TO RECEIVE COMPLETION CODE
048C 1074 : IERROR(AP) ADDRESS OF LONGWORD TO RECEIVE SECOND
048C 1075 : LONGWORD OF I/O STATUS BLOCK IF COMPLETION
048C 1076 : CODE IS SS$_DEVCMERR, SS$_DEVREQERR, OR
048C 1077 : SS$_CTRLERR
048C 1078 :
048C 1079 : IMPLICIT INPUTS:
048C 1080 : NONE
048C 1081 :
048C 1082 : OUTPUT PARAMETERS:
048C 1083 :
048C 1084 : ISTAT(AP) SEE ABOVE
048C 1085 : IERROR(AP) SEE ABOVE
048C 1086 :
048C 1087 : IMPLICIT OUTPUTS:
048C 1088 : NONE
048C 1089 :
048C 1090 : COMPLETION CODES:
048C 1091 :
048C 1092 : VARIOUS SYSTEM STATUS'S
048C 1093 :
048C 1094 : SIDE EFFECTS:
048C 1095 : NONE
048C 1096 :
048C 1097 : --
048C 1098 :
048C 1099 :
048C 1100 :
003C 048C 1101 .ENTRY LPASLOADMC, ^M<R2,R3,R4,R5>
048E 1102
54 6C 9A 048E 1103 MOVZBL (AP),R4 ; GET NUMBER OF ARGUMENTS
55 53 D4 0491 1104 CLRL R3 ; DEFAULT NUMBER TO APPEND TO LOG. NAME
01 00 0493 1105 MOVL #1,R5 ; DEFAULT MICROCODE TYPE
0496 1106
0496 1107 ; PROCESS ITYPE

```

Sym  
--  
DEL  
DEL  
DEL  
DEQ  
ENO  
ENT  
EXE  
EXE  
EXE  
EXE  
EXE  
EXE  
EXE  
EXE  
FET  
FET  
FET  
FIN  
FIN  
FIN  
FIN  
FIN  
FIN  
FIN  
FIN  
FIN  
FIN  
FIN  
FIN  
FIN  
FIN  
FLU  
FMG  
GET  
INI  
INI  
JOB  
JOB  
JOB  
LIB  
LIB  
LOC  
LOC  
NL  
OPA  
OPE  
PAL  
PAL

LPASLOADMC - LOAD MICROCODE

```

54 D7 0496 1108 DECL R4 ; ENOUGH ARGUMENTS SUPPLIED?
16 19 0498 1109 BLSS 50$ ; NO
50 04 AC D0 049A 1110 MOVL 4(AP),R0 ; GET ADDRESS OF ITYPE
03 13 049E 1111 BEQL 40$ ; DEFAULTED
55 60 9A 04A0 1112 MOVZBL (R0),R5 ; GET ITYPE
04A3 1113
04A3 1114 40$: ; PROCESS INUM
54 D7 04A3 1115 DECL R4 ; ENOUGH ARGUMENTS SUPPLIED?
09 19 04A5 1116 BLSS 50$ ; NO
50 08 AC D0 04A7 1117 MOVL 8(AP),R0 ; GET ADDRESS OF INUM
03 13 04AB 1118 BEQL 50$ ; DEFAULTED
53 60 3C 04AD 1119 MOVZWL (R0),R3 ; GET INUM
04B0 1120
04B0 1121 50$: ; ASSIGN CHANNEL TO LPA-11
52 7E 3E 04B0 1122 MOVAW -(SP),R2 ; GET ADDRESS OF WORD ON TOP OF
04B3 1123 ; STACK TO RECEIVE CHANNEL NUMBER
0029 30 04B3 1124 BSBW LPASS$ASSIGN ; ASSIGN CHANNEL
0B 50 E9 04B6 1125 BLBC R0,70$ ; ERROR
04B9 1126
04B9 1127 ; NOW SEND REQUEST TO LOADER PROCESS
55 DD 04B9 1128 PUSHL R5 ; PUSH MICROCODE TYPE
62 3F 04BB 1129 PUSHAW (R2) ; PUSH ADDRESS OF CHANNEL
00000000'EF 02 FB 04BD 1130 CALLS #2,LPASS$SNDLDRQ ; SEND LOAD REQUEST
04C4 1131
04C4 1132 70$: ; PROCESS ISTAT
54 D7 04C4 1133 DECL R4 ; ENOUGH ARGUMENTS SUPPLIED?
16 19 04C6 1134 BLSS 90$ ; NO
52 0C AC D0 04C8 1135 MOVL 12(AP),R2 ; GET ADDRESS OF ISTAT
03 13 04CC 1136 BEQL 80$ ; DEFAULTED
62 50 D0 04CE 1137 MOVL R0,(R2) ; STORE ISTAT
04D1 1138
04D1 1139 80$: ; PROCESS IERROR
54 D7 04D1 1140 DECL R4 ; ENOUGH ARGUMENTS SUPPLIED?
09 19 04D3 1141 BLSS 90$ ; NO
52 10 AC D0 04D5 1142 MOVL 16(AP),R2 ; GET ADDRESS OF IERROR
03 13 04D9 1143 BEQL 90$ ; DEFAULTED
62 51 D0 04DB 1144 MOVL R1,(R2) ; STORE IERROR
04DE 1145
04 04DE 1146 90$: RET

```

-\$25  
Symt  
---  
PROG  
PROG  
QUEL  
REAL  
RELE  
REMC  
REQ  
RESE  
RESE  
RESL  
RESL  
REWF  
SCAN  
SCH1  
SCH1  
SCH1  
SCH1  
SCHE  
SCSI  
SEAF  
SENC  
SENC  
SGN1  
SIGN  
SJC  
SND1  
SND1  
SND1  
STAF  
STAF  
STAF  
STAF  
STAF  
STOF  
STOF  
STOF  
SYME  
SYME  
SYSI

```

04DF 1148 .SBTTL LPASS$ASSIGN - ASSIGN A CHANNEL TO AN LPA-11
04DF 1149 :++
04DF 1150 : FUNCTIONAL DESCRIPTION:
04DF 1151 :
04DF 1152 : THIS ROUTINE ASSIGNS A CHANNEL TO A LOGICAL NAME OF THE FORM
04DF 1153 : LPA11$n, WHERE n IS THE VALUE OF ONE OF THE INPUT ARGUMENTS.
04DF 1154 : IT IS ASSUMED THAT THE USER HAS ALREADY ASSIGNED THAT LOGICAL NAME
04DF 1155 : TO AN LPA-11.
04DF 1156 :
04DF 1157 : CALLING SEQUENCE:
04DF 1158 :
04DF 1159 : BSBW/B
04DF 1160 :
04DF 1161 : INPUT PARAMETERS:
04DF 1162 :
04DF 1163 : R2 IS THE ADDRESS OF THE LOCATION TO STORE THE CHANNEL NUMBER
04DF 1164 : ASSIGNED
04DF 1165 : R3 CONTAINS THE NUMBER TO BE APPENDED TO THE LOGICAL NAME
04DF 1166 : TO ASSIGN A CHANNEL TO.
04DF 1167 :
04DF 1168 : IMPLICIT INPUTS:
04DF 1169 :
04DF 1170 : NONE
04DF 1171 :
04DF 1172 : OUTPUT PARAMETERS:
04DF 1173 :
04DF 1174 : R0 CONTAINS A COMPLETION CODE
04DF 1175 :
04DF 1176 : IMPLICIT OUTPUTS:
04DF 1177 :
04DF 1178 : NONE
04DF 1179 :
04DF 1180 : COMPLETION CODES:
04DF 1181 :
04DF 1182 : THE SAME ONES THAT ARE SUPPLIED BY THE $ASSIGN SYSTEM SERVICE
04DF 1183 :
04DF 1184 : SIDE EFFECTS:
04DF 1185 :
04DF 1186 : R1 IS NOT PRESERVED
04DF 1187 :
04DF 1188 :--
04DF 1189 :
04DF 1190 LPASS$ASSIGN:
54 53 53 54 04 04E4 1191 3C BB 04DF 1191 PUSHR #^M<R2,R3,R4,R5> ; SAVE SOME REGISTERS
7E 54 30 89 04E6 1192 04E1 1192 ;
54 53 53 54 04 04E4 1193 55 5E D0 04E1 1193 MOVL SP,R5 ; SAVE STACK POINTER
04E4 1194 ; CONVERT NUMBER TO ASCII STRING ON STACK
04E4 1195 ;
04E4 1196 ;
04E4 1197 10$: ; CONVERT NEXT DIGIT
04E4 1198 CLRL R4 ; HIGH BITS OF DIVIDEND
04E6 1199 EDIV #10,R3,R3,R4 ; QUO.-> R3 REM. -> R4
04EB 1200 BISB3 #^X$0,R4,-(SP) ; CONVERT TO ASCII AND PUSH ON STACK
04EF 1201 TSTL R3 ; REPEAT?
04F1 1202 BNEQ 10$ ; BR. IF YES
04F3 1203 ;
04F3 1204 ; NOW PUSH PREFIX STRING ONTO STACK

```

LPASSASSIGN - ASSIGN A CHANNEL TO AN LPA

```

7E   53   05   D0   04F3  1205      MOVL   #DNPREFIXS-1,R3      ; LENGTH OF STRING
      FB05  CF43  90   04F6  1206  20$:  MOVB   DNPREFIX[R3],-(SP)   ; PUSH NEXT CHAR ON STACK
      F7 53   F4   04FC  1207      SOBGEQ  R3,20$             ; REPEAT
                                04FF  1208
                                04FF  1209      ; NOW BUILD A STRING DESCRIPTOR ON STACK
53   55   5E   C3   04FF  1210      SUBL3  SP,R5,R3           ; OVERALL LENGTH OF STRING
      6E   9F   0503  1211      PUSHAB (SP)              ; PUSH ADDRESS OF STRING
      53   DD   0505  1212      PUSHL  R3                ; PUSH LENGTH
54   5E   D0   0507  1213      MOVL   SP,R4             ; R4 POINTS TO STRING DESCRIPTOR
                                050A  1214
                                050A  1215      ; NOW ASSIGN THE CHANNEL
                                050A  1216      $ASSIGN_S (R4),(R2)
      SE   55   D0   0517  1218      MOVL   R5,SP             ; RESTORE STACK POINTER
      3C   BA   051A  1219      POPR   #^M<R2,R3,R4,R5>  ; RESTORE REGISTERS
      05   051C  1220      RSB                                ; RETURN CODE IN R0 FROM $ASSIGN
                                051D  1221
                                051D  1222
                                051D  1223
                                051D  1224
                                .END

```







