

2 C DEPRESS A COMPRESSED DECK, APPLYING CORRECTIONS FROM DI.
 3 C
 4 C CHANGE CONTROL CARDS HAVE THE FORM
 5 C -M TO INDICATE THAT TEXT FROM SI FOLLOWING THE CHANGE CONTROL
 6 C CARD ARE TO BE INSERTED FOLLOWING LINE M OF THE PI TEXT STREAM, OR
 7 C -M,N TO INDICATE THAT TEXT FROM SI FOLLOWING THE CHANGE CONTROL
 8 C CARD ARE TO REPLACE LINES M THROUGH N OF THE PI TEXT STREAM.
 9 C

10 C THE CHANGE CONTROL CARD SIGNAL CHARACTER MAY BE CHANGED BY USING
 11 C ==* WHERE * MAY BE ANY CHARACTER. THE * THEN ASSUMES THE
 12 C ROLE OF THE DASH.
 13 C

14 C THE SEQUENCE -- ~~IS INTERPRETED AS AN END OF FILE MARK.~~ MEANS END OF CORRECTIONS,
 15 C IF COLUMN 3 IS ALSO EQUAL -, READ ANOTHER DECK ELSE STOP

16 INTEGER DIGITS(10)
 17 INTEGER DASH,COMMA,EQUAL,BLANK
 18 INTEGER NBR1,NBR2,NBRNBR,PINBR
 19 INTEGER PI,DI,SO,SS,PO
 20 INTEGER PIBFR(80),DIBFR(80)
 21 INTEGER NBR(2)
 22 EQUIVALENCE (NBR(1),NBR1), (NBR(2),NBR2)
 23 EXTERNAL READCB
 24 DATA DIGITS /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
 25 DATA DASH,COMMA,EQUAL,BLANK /1H-,1H,,1H=,1H /
 26 ~~DATA PINBR /0/~~
 27 DATA PI /4/, DI /11/, SO /3/, SS /8/, PO /10/
 28 ~~DATA MAXSI /0/~~

29 C
 30 C COPY CORRECTION CARDS FROM DI TO SS.
 31 C THREE ITEMS PRECEED EVERY RECORD ON SS: AN INDICATOR OF THE
 32 C RECORD TYPE, WHICH IS 1 OR 2 FOR A CORRECTION RECORD, OR 3 FOR
 33 C A TEXT RECORD, AND THE TWO CORRECTION NUMBERS (IF ANY).
 34 C

35 ① REWIND SS
 36 10 READ (DI,20) DIBFR
 37 20 FORMAT (80A1)
 38 IF (IOCHK(IOSTAT).EQ.0) GO TO 50
 39 C IF EOF OR ERROR, SET CORRECTION LINE NUMBER TO INFINITY
 40 30 NBR1=32767
 41 NBRNBR=1
 42 WRITE (SS,40) NBRNBR,NBR,DIBFR
 43 40 FORMAT (3A2,80A1)
 44 END FILE SS
 45 REWIND SS
 46 C SETUP DECOMPRESSION AND OUTPUT ROUTINES.
 47 CALL DEPRES (0,0,0,READCB)
 48 CALL OUTPUT (0,0)
 49 GO TO 220
 50 50 IF (DIBFR(1).EQ.DASH) GO TO 70
 51 C COPY DI TO SS IF NOT A DASH CARD.
 52 NBRNBR=3
 53 60 WRITE (SS,40) NBRNBR,NBR,DIBFR
 54 GO TO 10
 55 70 IF (DIBFR(2).NE.EQUAL) GO TO 80
 56 C CHANGE CONTROL CHARACTER
 57 DASH=DIBFR(3)
 58 GO TO 10

INTEGER ENDFLG

INTEGER SIGNAL

ENDFLG=BLANK

GO TO 30
 25 ENDFLG=DIBFR(3)

SIGNAL

SIGNAL

```

59 80 IF (DIBFR(2).EQ.DASHSIGNAL) GO TO 30 25
60 NBR1=0
61 NBRNBR=1
62 ICOL=2
63 90 IF (DIBFR(ICOL).EQ.COMMA) GO TO 140
64 IF (DIBFR(ICOL).EQ.BLANK) GO TO 190
65 DO 100 J=1,10
66 IF (DIBFR(ICOL).EQ.DIGITS(J)) GO TO 130
67 100 CONTINUE
68 110 WRITE (SO,120) (DIBFR(I),I=1,40)
69 120 FORMAT (2X,40A1/2X,28HCORRECTION CARD FORMAT ERROR)
70 STOP
71 130 NBR1=10*NBR1+J-1
72 ICOL=ICOL+1
73 GO TO 90
74 140 NBR2=0
75 NBRNBR=2
76 150 ICOL=ICOL+1
77 IF (DIBFR(ICOL).EQ.BLANK) GO TO 180
78 DO 160 J=1,10
79 IF (DIBFR(ICOL).EQ.DIGITS(J)) GO TO 170
80 160 CONTINUE
81 GO TO 110
82 170 NBR2=10*NBR2+J-1
83 GO TO 150
84 180 IF (NBR1.GT.NBR2) GO TO 110
85 190 IF (NBR1.GE.MAXSI) GO TO 210
86 WRITE (SO,200) (DIBFR(I),I=1,40)
87 200 FORMAT (2X,40A1/2X,30HCORRECTION CARD SEQUENCE ERROR)
88 STOP
89 210 MAXSI=NBR(NBRNBR)
90 GO TO 60
91 C
92 C READ CORRECTION FILE AND TAKE APPROPRIATE ACTION.
93 C
94 220 READ (SS,40) NBRNBR,NBR,DIBFR
95 GO TO (240,240,230), NBRNBR
96 C
97 C COPY TEXT TO PO.
98 C
99 230 CALL OUTPUT (DIBFR,1104)
100 GO TO 220
101 C
102 C DECOMPRESS IMAGES, WRITING ONTO PO UNTIL NBR1' TH IMAGE OUTPUT.
103 C
104 240 IF (PINBR.GE.NBR1) GO TO (220,330), NBRNBR
105 IFLAG=1
106 DO 250 I=1,80
107 250 PIBFR(I)=0
108 CALL DEPRES (PIBFR,IFLAG,0,READCB)
109 IF (IFLAG.NE.0) GO TO 310
110 260 IF (NBR1.NE.32767) GO TO 290
111 C
112 C CLOSE INPUT AND OUTPUT ROUTINES
113 C
114 280 CALL DEPRES (0,-1,0,READCB)
115 CALL OUTPUT (0,-1)
116 CALL EXIT

```

IF (ENDFLG.EQ.SIGNAL) GO TO 1

(END FILE PD

PAGE 4 06/25/77 DEPUPD VORTXII FTN IV(G) DEPUPD 1107 HOURS

```
117 290 WRITE (SO,300)
118 300 FORMAT (22H PI EOF BEFORE DI EOF)
119 GO TO 280
120 310 PINBR=PINBR+1
121 IF (NBRNBR.EQ.1) GO TO 320
122 C DO NOT COPY THE NBR1' TH IMAGE IF REPLACING.
123 IF (PINBR.GE.NBR1) GO TO 330
124 320 CALL OUTPUT (PIBFR,80)
125 GO TO 240
126 C
127 C SKIP PI UNTIL NBR2' TH RECORD SKIPPED.
128 C
129 330 IF (PINBR.GE.NBR2) GO TO 220
130 IFLAG=1
131 DO 340 I=1,80
132 340 PIBFR(I)=0
133 CALL DEPRES (PIBFR,IFLAG,0,READCB)
134 IF (IFLAG.EQ.0) GO TO 260
135 PINBR=PINBR+1
136 GO TO 330
137 C
138 END
0 ERRORS COMPILATION COMPLETE
```

1 TITLE DEPRES

```

2      SUBROUTINE DEPRES (IMAGE,FLAG,INSERT,GETBLK)
3      C
4      C      DE-PRESS A COMPRESSED IMAGE, RE-INSERTING THE CHARACTER 'INSERT'.
5      C
6      C      THE OUTPUT IS RECONSTRUCTED IN IMAGE.
7      C
8      C      FLAG=0 MEANS OPEN INPUT FILE,
9      C      FLAG<0 MEANS CLOSE INPUT FILE,
10     C      FLAG>0 MEANS RECONSTRUCT AN IMAGE.
11     C
12     C      ON RETURN FLAG WILL BE SET >0 IF AN IMAGE WAS DECODED,
13     C              OR =0 IF END-OF-FILE WAS DETECTED.
14     C
15     C      THE SUBROUTINE GETBLK WILL BE USED TO READ INPUT.
16     C
17     C      INTEGER IMAGE(1),FLAG,INSERT,INPBUF(156),BUFLEN,GC,RC,SC,CPIB
18     C      DATA BUFLN/156/
19     C
20     C      DECIDE WHAT TO DO BASED UPON FLAG.
21     C
22     C      IF (FLAG) 110,100,10
23     C
24     C      PROVIDE AN IMAGE.
25     C
26     10  IF (CPIB.LT.BUFLN) GO TO 20
27     CALL GETBLK (INPBUF,1)
28     CPIB=0
29     20  CPIB=CPIB+1
30     C      GET THE GROUP COUNTER.
31     GC=INPBUF(CPIB)
32     FLAG=GC
33     IF (GC.EQ.0) GO TO 120
34     IMGLN=0
35     C      UNSCRAMBLE GC GROUPS.
36     DO 90 I=1,GC
37     IF (CPIB.LT.BUFLN) GO TO 30
38     CALL GETBLK (INPBUF,1)
39     CPIB=0
40     30  CPIB=CPIB+1
41     C      GET THE REMOVED CHARACTER COUNT.
42     RC=INPBUF(CPIB)
43     IF (RC.EQ.0) GO TO 50
44     C      RE-INSERT RC OCCURRENCES OF 'INSERT'.
45     DO 40 J=1,RC
46     IMGLN=IMGLN+1
47     40  IMAGE(IMGLN)=INSERT
48     50  IF (CPIB.LT.BUFLN) GO TO 60
49     CALL GETBLK (INPBUF,1)
50     CPIB=0
51     60  CPIB=CPIB+1
52     C      GET SIGNIFICANT CHARACTER COUNT.
53     SC=INPBUF(CPIB)
54     IF (SC.EQ.0) GO TO 90
55     C      COPY SC CHARACTERS FROM INPBUF TO IMAGE.
56     DO 80 J=1,SC
57     IF (CPIB.LT.BUFLN) GO TO 70
58     CALL GETBLK (INPBUF,1)

```

```
59      CPIB=0
60      70      CPIB=CPIB+1
61      IMGLN=IMGLN+1
62      80      IMAGE(IMGLN)=INPBUF(CPIB)
63      90      CONTINUE
64      GO TO 120
65      C
66      C      OPEN THE INPUT FILE.
67      C
68      100     CALL GETBLK (0,0)
69      CPIB=BUFLN
70      GO TO 120
71      C
72      C      CLOSE THE INPUT FILE.
73      C
74      110     CALL GETBLK (0,-1)
75      120     RETURN
76      END
0 ERRORS. COMPILATION COMPLETE
```


1 TITLE OUTPUT

```

2      SUBROUTINE OUTPUT (IMAGE,IFLAG)
3      C
4      C      OUTPUT THE IMAGE TO THE PROCESSOR OUTPUT (PO) UNIT.
5      C
6      C      IFLAG=0 MEANS OPEN OUTPUT.
7      C      IFLAG>1024 MEANS OUTPUT IFLAG-1024 CHARACTERS, ONE PER WORD,
8      C      WITHOUT TRANSLATION.
9      C      0<IFLAG<1025 MEANS OUTPUT IFLAG CHARACTERS, ONE PER WORD.
10     C      IFLAG<0 MEANS CLOSE OUTPUT.
11     C
12     INTEGER BLANK,DEVTP,DISK,IMAGE(1),PO,POSN,WORK(240),XLATE(64)
13     DATA BLANK /2H /, DISK /2HD /, PO /10/
14     C      0 1 2 3 4 5 6 7
15     DATA XLATE /
16     1 1H ,1H!,1H",1H#,1H$,1H%,1H&,1H',
17     2 1H(,1H),1H*,1H+,1H-,1H.,1H/,
18     3 1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,
19     4 1H8,1H9,1H:,1H;,1H<,1H=,1H>,1H?,
20     5 1H@,1H^,1H^,1H^,1H^,1H^,1H^,1H^,
21     6 1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,
22     7 1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,
23     8 1HX,1HY,1HZ,1H[,1H\,1H],1H<,1H↑/
24     C
25     IF (IFLAG) 10,50,60
26     C      CLOSE OUTPUT
27     10  IF (POSN.EQ.0) GO TO 40 160
28     IF (POSN.GE.240) GO TO 30
29     POSN=POSN+1
30     DO 20 I=POSN,240
31     20  WORK(I)=BLANK
32     30  WRITE (PO,120) WORK
33     40  END FILE PO
34     GO TO 160
35     C      OPEN OUTPUT
36     50  CALL DEVICE (PO,DEVTP)
37     POSN=0
38     GO TO 160
39     C      WRITE IFLAG CHARACTERS TO PO.
40     60  K=0
41     NCHAR=IFLAG
42     IF (NCHAR.LE.1024) GO TO 70
43     K=1
44     NCHAR=IFLAG-1024
45     70  IF (DEVTP.EQ.DISK) GO TO 110
46     C      CONVERT CHOPPED ASCII TO INTERNAL ASCII.
47     DO 90 I=1,NCHAR
48     IF (K.EQ.0) GO TO 80
49     WORK(I)=IMAGE(I)
50     GO TO 90
51     80  J=IMAGE(I)+1
52     WORK(I)=XLATE(J)
53     90  CONTINUE
54     WRITE (PO,100) (WORK(I),I=1,NCHAR)
55     100 FORMAT (80A1)
56     GO TO 160
57     110 DO 150 I=1,NCHAR
58     J=IMAGE(I)+1

```



```
59 IF (POSN.LT.240) GO TO 130
60 WRITE (PO,120) WORK
61 120 FORMAT (240A1)
62 POSN=0
63 130 POSN=POSN+1
64 IF (K.NE.0) GO TO 140
65 WORK(POSN)=XLATE(J)
66 GO TO 150
67 140 WORK(POSN)=IMAGE(I)
68 150 CONTINUE
69 160 RETURN
70 C
```

```
71 END
0 ERRORS COMPILATION COMPLETE
/MEM,10
/DASMR
```

000002	A	19	SI	EQU	2	
000003	A	20	SO	EQU	3	
000004	A	21	PI	EQU	4	
000400	A	22	VSLUT1	EQU	0400	LUT BASE
000355	A	23	VSDSTB	EQU	0355	DEVICE SPECIFICATION TABLE

	24	EJEC		
	25	* EXAMINE FLAG TO DETERMINE WHAT TO DO		
000000	024170	A	26 READC1	LDB LFLG
000001	016000	A	27	LDA 0,2
000002	001010	A	28	JAZ OPN OPEN INPUT
000003	000146	R		
000004	001004	A	29	JAN RETURN CLOSE INPUT (NOTHING TO DO).
000005	000163	R		
000006	001000	A	30	JMP RDCD
000007	000034	R		
	31	* BACKSPACE PI TO RE-READ LAST RECORD - R RESPONSE TO ERROR MSG.		
000010	014470	A	32 REREAD	LDA DEVPI DEVICE TYPE FOR PI
000011	130000	L	33	ERA ='M ' IS IT MAG TAPE?
000012	001016	A	34	JANZ RDSYNC NO, SKIP BACKSPACE
000013	000023	R		
	35	SREC PIDCB,PI,0,1 BACKSPACE PI		
000014	006505	A		
000015	000000	E		
000016	100000	A		
000017	012004	A		
000020	000355	R		
000021	000000	A		
000022	000000	A		
	36	* READ A CARD AND UNPACK IT.		
	37	RDSYNC READ PIDCB,PI,0,3 READ PI UNFORMATTED SYNCHRONOUS		
000023	006505	A		
000024	000015	E		
000025	100000	A		
000026	030004	A		
000027	000355	R		
000030	000000	A		
000031	000000	A		
000032	001000	A	38	JMP GOTCRD
000033	000043	R		
	39	RDCD STAT SEEK,IOERR,EOF,EOF,RDCD WAIT FOR COMPLETION		
000034	006505	A		
000035	000000	E		
000036	000154	R		
000037	000174	R		
000040	000177	R		
000041	000177	R		
000042	000034	R		
000043	014314	A	40 GOTCRD	LDA BUFI COLUMN 1
000044	006150	A	41	ANAI 03000 ROWS 11,0
000045	003000	A		
000046	006130	A	42	ERA1 03000 ROWS 11,0 BOTH SET?
000047	003000	A		
000050	001010	A	43	JAZ CB YES, CARD IS COLUMN BINARY
000051	000055	R		
000052	002000	A	44	CALL ERRMSG,1 PRINT MESSAGE, RETURN TO RDCD
000053	000202	R		
000054	000001	A		
000055	014302	A	45 CB	LDA BUFI COLUMN 1
000056	006150	A	46	ANAI 0400 ROW 1 SET?
000057	000400	A		
000060	001010	A	47	JAZ NCKS NO CHECKSUM IF ROW 1 NOT SET
000061	000100	R		
000062	014276	A	48	LDA BUFI+1 COLUMN 2 = CHECKSUM

000063	024031	A	49	LDB	BUFL	
000064	006136	A	50	CKLP	BUFI+2,2	ACCUMULATE CHECKSUM
000065	000362	R				
000066	001020	A	51	JBZ	CKLX	
000067	000073	R				
000070	005322	A	52	DBR		
000071	001000	A	53	JMP	CKLP	
000072	000064	R				
000073	001010	A	54	CKLX	JAZ	NCKS
000074	000100	R				JUMP IF CHECKSUM OK
000075	002000	A	55	CALL	ERRMSG,2	PRINT MESSAGE, RETURN TO RDCD
000076	000202	R				
000077	000002	A				
000100	014377	A	56	NCKS	LDA	SEQ
000101	005111	A	57		IAR	
000102	134255	A	58		ERA	BUFI
000103	006150	A	59		ANAI	0377
000104	000377	A				L.O. BITS OF SEQ+1 EQUAL ROWS 2-9?
000105	001010	A	60	JAZ	SEQOK	YES - SEQOK
000106	000112	R				
000107	002000	A	61	CALL	ERRMSG,3	PRINT MESSAGE, RETURN TO READCD
000110	000202	R				
000111	000003	A				
000112	044365	A	62	SEQOK	INR	SEQ
000113	034054	A	63		LDX	LINP
000114	006020	A	64		LOBI	77
000115	000115	A				BUMP SEQUENCE NUMBER
000115			65	BUFL	BES	0
			66	*	MOVE	DATA TO USERS AREA - NOTE DATA IS MOVED BACKWARD.
000116	006016	A	67	MOVL	LDAE	BUFI+2,2
000117	000362	R				GET WORD OF CARD
000120	154006	A	68		ANA	K077
000121	055233	A	69		STA	155,1
000122	005344	A	70		DXR	
000123	006016	A	71		LDAE	BUFI+2,2
000124	000362	R				GET WORD OF CARD
000125	004346	A	72		LSRA	6
000126	006150	A	73		ANAI	077
000127	000077	A				GET ROWS 12-3
000127			74	K077	BES	0
000130	055233	A	75		STA	155,1
000131	001020	A	76		JBZ	CKEOF
000132	000137	R				EXIT IF FINISHED
000133	005322	A	77		DBR	
000134	005344	A	78		DXR	
000135	001000	A	79		JMP	MOVL
000136	000116	R				
000137	014220	A	80	CKEOF	LDA	BUFI
000140	006150	A	81		ANAI	04000
000141	004000	A				COLUMN 1
000142	001010	A	82	JAZ	SEEK	ROW 12
000143	000154	R				NOT SET, GO START I/O FOR NEW CARD.
000144	001000	A	83	JMP	RETURN	
000145	000163	R				
			84	*	OPEN INPUT	(ASSUME IT IS NOT RMD, USE PI LOCAL DCB).
000146	005001	A	85	OPN	TZA	
000147	054330	A	86		STA	SEQ
000150	002000	A	87	CALL	DEVICE,=PI,DEVPI	CLEAR SEQUENCE NUMBER

000151 000527 R
000152 000000 L
000153 000501 R

88 SEEK READ PIDCB,PI,1,3 READ PI UNFORMATTED ASYNCHRONOUSLY

000154 006505 A
000155 000024 E
000156 100000 A
000157 130004 A
000160 000355 R
000161 000000 A
000162 000000 A
000163 001000 A
000164 000000 A

89 RETURN JMP *-*

	90	EJEC		
	91 *	ENTRY AND EXIT		
000164	92 READCB	BES	0	
000165 002000 A	93	CALL	\$SE,2	PROCESS CALLING SEQUENCE
000166 000000 E				
000167 000002 A				
000170 000000 A	94 LINP	DATA	0	LOCATION OF INPUT BUFFER
000171 000000 A	95 LFLG	DATA	0	LOCATION OF FLAG
000172 001000 A	96	JMP	READC1	START
000173 000000 R				

```

          97      EJEC
          98 * PRINT ERROR MESSAGE
          99 IOERR CALL ERRMSG,4      I/O ERROR ON PI

000174 002000 A
000175 000202 R
000176 000004 A
000177 002000 A 100 EOF CALL ERRMSG,5      UNEXPECTED EOF ON PI
000200 000202 R
000201 000005 A
000202 000000 A 101 ERRMSG ENTR
000203 006037 A 102 LDxE* ERRMSG      GET ERROR MESSAGE NUMBER
000204 100202 R
000205 006015 A 103 LDAE MSGA,1
000206 000344 R
000207 006145 A 104 SUBE MSGA-1,1      COMPUTE MESSAGE LENGTH
000210 000343 R
000211 054140 A 105 STA SODCB
000212 006015 A 106 LDAE MSGA-1,1
000213 000343 R
000214 054136 A 107 STA SODCB+1      STORE MESSAGE ADDRESS
          108 PMSG WRITE SODCB,SO,0,1 PRINT MESSAGE

000215 006505 A
000216 000155 E
000217 100000 A
000220 010403 A
000221 000352 R
000222 000000 A
000223 000000 A
          109      READ PIDCB,SO,0,1      READ RESPONSE FROM SO INTO BUFI

000224 006505 A
000225 000216 E
000226 100000 A
000227 010003 A
000230 000355 R
000231 000000 A
000232 000000 A
000233 014124 A 110 LDA BUFI      RESPONSE
000234 004350 A 111 LSRA 8      RIGHT JUSTIFY
000235 006140 A 112 SUBI 'R'
000236 000322 A
000237 001010 A 113 JAZ REREAD      RE-READ IMAGE IF R RESPONSE
000240 000010 R
000241 005311 A 114 DAR
000242 001010 A 115 JAZ RDSYNC      READ NEXT IF SKIP RESPONSE
000243 000023 R
000244 006140 A 116 SUBI 'X'-'S'
000245 000005 A
000246 002010 A 117 JAZM EXIT      CALL EXIT IF X RESPONSE
000247 000000 E
000250 001000 A 118 JMP PMSG      PRINT MESSAGE AGAIN IF WRONG RESPONSE
000251 000215 R

119 * MESSAGE TABLE. MESSAGES MUST BE CONTIGUOUS AND IN ORDER.
120 MSG1 DATA 'CARD NOT BINARY - RSX'

000252 120303 A
000253 140722 A
000254 142240 A
000255 147317 A
000256 152240 A
000257 141311 A
000260 147301 A

```


000261 151331 A
 000262 120255 A
 000263 120322 A
 000264 151730 A
 000265 120303 A
 000266 144305 A
 000267 141713 A
 000270 151725 A
 000271 146640 A
 000272 142722 A
 000273 151317 A
 000274 151240 A
 000275 126640 A
 000276 151323 A
 000277 154240 A
 000300 120323 A
 000301 142721 A
 000302 152705 A
 000303 147303 A
 000304 142640 A
 000305 142722 A
 000306 151317 A
 000307 151240 A
 000310 126640 A
 000311 151323 A
 000312 154240 A
 000313 120311 A
 000314 127717 A
 000315 120305 A
 000316 151322 A
 000317 147722 A
 000320 120317 A
 000321 147240 A
 000322 150311 A
 000323 120255 A
 000324 120322 A
 000325 151730 A
 000326 120325 A
 000327 147305 A
 000330 154320 A
 000331 142703 A
 000332 152305 A
 000333 142240 A
 000334 142717 A
 000335 143240 A
 000336 147716 A
 000337 120320 A
 000340 144640 A
 000341 126640 A
 000342 151323 A
 000343 154240 A

121 MSG2 DATA ' CHECKSUM ERROR - RSX '

122 MSG3 DATA ' SEQUENCE ERROR - RSX '

123 MSG4 DATA ' I/O ERROR ON PI - RSX'

124 MSG5 DATA ' UNEXPECTED EOF ON PI - RSX '

125 * MESSAGE ADDRESS TABLE MUST IMMEDIATELY FOLLOW MESSAGES.

000344 000252 R 126 MSGA DATA MSG1,MSG2,MSG3,MSG4,MSG5

000345 000265 R

000346 000300 R

000347 000313 R

000350 000326 R

000351 000344 R

127 DATA MSGA USED TO COMPUTE LENGTH OF LAST MESSAGE

128 *

129 *

DATA AREAS

130 SODCB DCB 0,0,0

DATA CONTROL BLOCK FOR SO

000352 000000 A

000353 000000 A

000354 000000 A

131 PIDCB DCB

80,BUFI,0

DATA CONTROL BLOCK FOR PI

000355 000120 A

000356 000360 R

000357 000000 A

000360

132 BUFI

BSS

80

INPUT BUFFER

000500 000000 A

133 SEQ

DATA

0

INPUT SEQUENCE NUMBER

000501 000000 A

134 DEVPI

DATA

0

DEVICE TYPE FOR PI

```

135      EJECT
136 *    GET DEVICE TYPE
137 *    CALL DEVICE (UNIT,DEVTP)
138 *    RETURNS FIRST LETTER OF DEVICE TYPE LJSF IN DEVTP.
139 *
140      NAME      DEVICE
000502 006027 A 141 DEV0  LDBE*  DEV#      GET FORTRAN UNIT NUMBER
000503 100533 R
000504 006216 A 142      LDAE*  V$LUT1,2,0200 GET LUT ENTRY
000505 100400 A
000506 006150 A 143      ANAI    0377      USE RIGHT HALFWORD
000507 000377 A
000510 005311 A 144      DAR
000511 054002 A 145      STA     *+3
000512 004201 A 146      ASLA    1
000513 006120 A 147      ADDI    *-*      MULTIPLY BY 3
000514 000000 A
000515 120355 A 148      ADD     V$DSTB
000516 005012 A 149      TAB
000517 016001 A 150      LDA     1,2      GET DST ENTRY
000520 006150 A 151      ANAI    0177400 GET LEFT HALFWORD
000521 177400 A
000522 006110 A 152      ORAI    0240      INCLUDE A SPACE IN RIGHT HALFWORD
000523 000240 A
000524 006057 A 153      STAE*  DEVTP    STORE DEVICE TYPE IN CALLING PROGRAM
000525 100534 R
000526 001000 A 154      JMP     *-*
000527 000000 A

155 *
000527      156 DEVICE BES      0      ENTRY AND EXIT
000530 002000 A 157      CALL    $SE,2    PROCESS CALLING SEQUENCE
000531 000166 E
000532 000002 A
000533 000000 A 158 DEV#  DATA    0      ADDRESS OF DEVICE # (FORTRAN UNIT)
000534 000000 A 159 DEVTP DATA    0      ADDRESS WHERE FORTRAN WANTS DEVICE TYPE
000535 001000 A 160      JMP     DEV0
000536 000502 R

```

161 END

ENTRY NAMES

000527 R DEVICE 000164 R READCB

EXTERNAL NAMES

000531 E \$SE 000247 E EXIT 000225 E V\$IOCB 000035 E V\$IOST

SYMBOLS

```

000531 E $SE 000360 R BUFI 000115 R BUFL 000055 R CB
000137 R CKEOF 000064 R CKLP 000073 R CKLX 000533 R DEV#
000502 R DEV0 000527 R DEVICE 000501 R DEVPI 000534 R DEVTP
000177 R EOF 000202 R ERRMSG 000247 E EXIT 000043 R GOTCRD
000174 R IOERR 000127 R K077 000171 R LFLG 000170 R LINP
000116 R MOVL 000252 R MSG1 000265 R MSG2 000300 R MSG3
000313 R MSG4 000326 R MSG5 000344 R MSGA 000100 R NCKS
000146 R OPN 000004 A PI 000355 R PIDCB 000215 R PMSG
000034 R RDCD 000023 R RDSYNC 000000 R READCB1 000164 R READCB
000010 R REREAD 000163 R RETURN 000154 R SEEK 000500 R SEQ
000112 R SEQOK 000002 A SI 000003 A SO 000352 R SODCB
000355 A V$DSTB 000225 E V$IOCB 000035 E V$IOST 000400 A V$LUT1

```

0 ERRORS ASSEMBLY COMPLETE

/LMGEN

1		1
2		2
3		3
4		4
5		5
6		6
7		7
8		8
9		9
10		10
11		11
12		12
13		13
14		14
15		15
16		16
17		17
18		18
19		19
20		20
21		21
22		22
23		23
24		24
25		25
26		26
27		27
28		28
29		29
30		30
31		31
32		32
33		33
34		34
35		35
36		36
37		37
38		38
39		39
40		40
41		41
42		42
43		43
44		44
45		45
46		46
47		47
48		48
49		49
50		50
51		51
52		52
53		53
54		54
55		55
56		56
57		57

TIDB,DEPUPD,1,0

LD,80,,80

LIR

!WCS0 A 55144	V\$LUT1 A 55110	V\$LUT2 A 55101	V\$LUT3 A 55100
V\$LUNT A 55027	!PD01B A 54731	!PD00B A 54632	V\$LRSK A 54311
V\$UTB A 54311	V\$PTVB A 54145	V\$FLRS A 54145	V\$TFC A 75777
V\$BFC A 75600	V\$GFCB A 75460	V\$JPBF A 75407	SSFCB A 75542
SIFCB A 75460	PIFCB A 75472	LOFCB A 75504	GOFCB A 75554
BOFCB A 75530	BIFCB A 75516	V\$RERS A 07073	\$PA A 06617
\$ER A 06614	V\$DSTB A 55144	POFCB A 75566	\$EE A 15612
WSX A 06164	V\$OPNB A 05751	V\$OPEN A 05760	V\$CLSB A 05536
V\$CLOS A 05577	R\$X A 06155	\$I6 A 06130	\$I5 A 06122
\$I4 A 06114	\$I3 A 06106	\$I2 A 06100	\$EQ A 05614
\$DQ A 05606	\$BA A 05500	V\$BIC A 73642	V\$ERR A 73230
V\$FNRM A 72705	IOFOOD A 72611	V\$FNR A 72411	V\$TBSR A 71460
V\$ALTB A 71400	V\$EXEC A 70243	V\$FNIS A 67652	V\$EMP A 67603
V\$IOST A 73552	V\$IOC A 71660	DEVICE A 05452	\$SE A 15603
V\$RER1 A 14205	V\$RERN A 00000	V\$RERR A 14205	EXIT A 07067
\$ST A 06623	\$DO A 06571	OUTPUT A 03475	DEPRES A 02613
\$EN A 05506	\$WR A 06054	IOCHK A 05555	\$ND A 05462
\$I1 A 06072	\$RD A 06046	\$RE A 05514	READCB A 05107
[\$IAP] A 00500	[\$LIT] A 00775	[\$PED] A 15771	

END,BL,E

/FINI

145,221,292
B-292
295-455

146-221

3.9
157