

digital  
digital  
digital  
digital  
digital  
digital  
digital

DMS11-F DEC/X11 MODULE  
(DMSA)  
DIAGNOSTIC

digital

**Computer Special Systems**



NOTEBOOK SECTION

Bernard COLLENOT  
CSS ANNECY DEC.79

OPTION NUMBER

DMS11-F DEC/X11 MODULE (DMSA)

DRAWING SET NUMBER

PROGRAM NUMBER

YP-Z00YD-X0

DOCUMENT NUMBER      REVISION

YP-Z00YE-X0

DATE

DECEMBER 1979


DMS11-F DEC/X11 MODULE  
(DMSA)  
DIAGNOSTIC

**digital**

**Computer Special Systems**

ANNECY

00000000000000000000

00000000000000000000

00000000000000000000

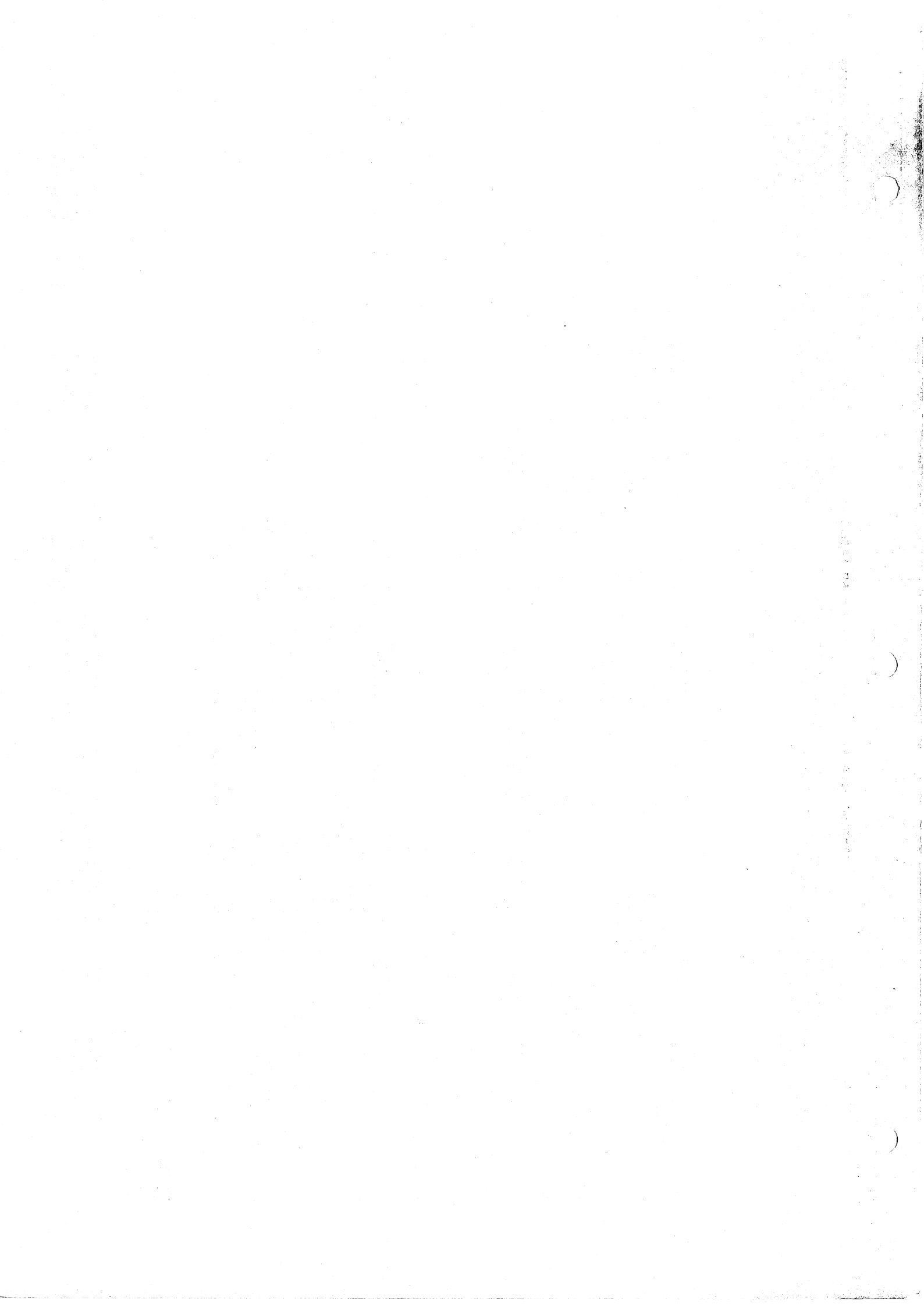
00000000000000000000

01234567890123456789	** RSX-11M-PLUS V1.0 **	[52,1]XDMSAO - NO PAGE LIMIT	1-JUL-80	16:46:56	01234567890123456789
01234567890123456789	** RSX-11M-PLUS V1.0 **	FORM #0 - NORMAL HARDWARE FORMS	1-JUL-80	16:46:56	01234567890123456789
01234567890123456789	** RSX-11M-PLUS V1.0 **	NO IMPLIED FORM FEED	1-JUL-80	16:46:56	01234567890123456789
01234567890123456789	** RSX-11M-PLUS V1.0 **	DB0:[52,2]XDMSAO.LST;1	1-JUL-80	16:46:56	01234567890123456789

CCCCCCCC	000000	LL	LL	EEEEEEEE	NN	NN	000000	TTTTTTTT
CCCCCCCC	000000	LL	LL	EEEEEEEE	NN	NN	000000	TTTTTTTT
CC	CC	LL	LL	EE	NN	NN	00 00	TT
CC	CC	LL	LL	EE	NN	NN	00 00	TT
CC	CC	LL	LL	EE	NNNN	NN	00 00	TT
CC	CC	LL	LL	EE	NNNN	NN	00 00	TT
CC	CC	LL	LL	EE	NN	NN	00 00	TT
CC	CC	LL	LL	EE	NN	NN	00 00	TT
CC	CC	LL	LL	EE	NN	NNNN	00 00	TT
CC	CC	LL	LL	EE	NN	NN	00 00	TT
CC	CC	LL	LL	EE	NN	NN	00 00	TT
CCCCCCCC	000000	LLLLLLLL	LLLLLLLL	EEEEEEEE	NN	NN	000000	TTTTTTTT
CCCCCCCC	000000	LLLLLLLL	LLLLLLLL	EEEEEEEE	NN	NN	000000	TTTTTTTT

XX	XX	DDDDDDDD	MM	MM	SSSSSSSS	AAAAAA	000000
XX	XX	DDDDDDDD	MM	MM	SSSSSSSS	AAAAAA	000000
XX	XX	DD	DD	MMMM	MMMM	SS	AA AA 00 00
XX	XX	DD	DD	MMMM	MMMM	SS	AA AA 00 00
XX	XX	DD	DD	MM	MM	SS	AA AA 00 0000
XX	XX	DD	DD	MM	MM	SS	AA AA 00 0000
XX	XX	DD	DD	MM	MM	SSSSSS	AA AA 00 00 00
XX	XX	DD	DD	MM	MM	SSSSSS	AA AA 00 00 00
XX	XX	DD	DD	MM	MM	SS	AAAAAAAAAA 0000 00
XX	XX	DD	DD	MM	MM	SS	AAAAAAAAAA 0000 00
XX	XX	DD	DD	MM	MM	SS	AA AA 00 00 00
XX	XX	DD	DD	MM	MM	SS	AA AA 00 00 00
XX	XX	DDDDDDDD	MM	MM	SSSSSSSS	AA	AA 000000
XX	XX	DDDDDDDD	MM	MM	SSSSSSSS	AA	AA 000000

01234567890123456789	** RSX-11M-PLUS V1.0 **	[52,1]XDMSAO - NO PAGE LIMIT	1-JUL-80	16:46:56	01234567890123456789
01234567890123456789	** RSX-11M-PLUS V1.0 **	FORM #0 - NORMAL HARDWARE FORMS	1-JUL-80	16:46:56	01234567890123456789
01234567890123456789	** RSX-11M-PLUS V1.0 **	NO IMPLIED FORM FEED	1-JUL-80	16:46:56	01234567890123456789
01234567890123456789	** RSX-11M-PLUS V1.0 **	DB0:[52,2]XDMSAO.LST;1	1-JUL-80	16:46:56	01234567890123456789



DMSA DEC/X11 SYSTEM EXERCISER MACRO M1113 01-JUL-80 16:42  
TABLE OF CONTENTS

8-	129	AUTOCONFIGURATION
9-	158	TEST 0
10-	236	TEST 1
11-	325	TEST 2
12-	461	END OF PASS
13-	467	AUTOMATIC LINE CONFIGURATION
14-	521	KMC CODE LOADER
15-	556	CLEAR KMC
16-	601	REQUEST CSR REGISTERS (RDYI INTEPPUP1)
17-	640	WAIT KMC ANSWER (RDYO INTERRUPT)
18-	685	RANDOM NUMBER GENERATOR SUBROUTINE
19-	703	ERROR MESSAGE FROM KMC
20-	734	SPECIAL COMMANDS
21-	770	ASCII STRINGS
23-	816	PROGRAM VARIABLES
24-	857	BUFFERS
37-	2	*KMC* **** DMS11F KMC MICROCODE ****
38-	74	*KMC* DEFINITIONS
39-	281	*KMC* INITIALISATION
40-	313	*KMC* MAIN LOOP
41-	334	*KMC* SERVICING L.U. REQUESTS
42-	414	*KMC* SCANNING ROUTINE
43-	445	*KMC* SERVICING OF CSR REGISTERS
44-	481	*KMC* SERVICING OF PDP REQUESTS
45-	560	*KMC* WT OUTPUT
46-	580	*KMC* SERVICING OF OUTPUT TO PDP
47-	690	*KMC* FIFO INPUT
49-	877	*KMC* ASK FOR STATUS
50-	899	*KMC* TABLE ADDRESS FOR RECEPTION
51-	928	*KMC* INVALID COMMAND
52-	941	*KMC* TX OF INFO FRAME
57-	1212	*KMC* TX DISPATCHING ROUTINE
58-	1243	*KMC* TX OF LONG CONTROL FRAME
63-	1369	*KMC* TX OF SHORT CONTROL FRAME
66-	1455	*KMC* LOAD BASE ADDRESS
67-	1491	*KMC* LOAD MASK
68-	1527	*KMC* ABORT TRANSMISSION
70-	1592	*KMC* ABORT RECEPTION
71-	1629	*KMC* MISCELLANEOUS COMMANDS
72-	1649	*KMC* RECEPTION ROUTINES
96-	2289	*KMC* RELAY ADDRESS IN PAGE 2
97-	2299	*KMC* FIFO OUTPUT READY
98-	2353	*KMC* ERROR STATUS
99-	2406	*KMC* MODEM STATUS RECEIVED
100-	2423	*KMC* COMMAND ACKNOWLEDGE
101-	2442	*KMC* FIFO TRACE
102-	2463	*KMC* QUEUING MESSAGES
109-	2703	*KMC* SUBROUTINES
112-	2792	*KMC* TRANSMISSION SUBROUTINES
118-	1	*KMC* BREAKPOINTS DEFINITION

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37

.REM-

PRODUCT CODE : YP-Z00YE-X0

ASSOCIATE PAPER TAPE : YP-Z00YD-X0

PRODUCT NAME: DMS11-F DEC/X11 MODULE (DMSA)

DATE: DECEMBER 1979

AUTHOR(S): BERNARD P. COLLENOT

MAINTAINER: C.S.S. ENGINEERING GROUP (ANNECY)

COPYRIGHT: 1979, DIGITAL EQUIPMENT CORP.  
MAYNARD MASS.



39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70

1. ABSTRACT

THE DMS DEC/X11 MODULE CONTROLS ONE DMS11-F  
LINE UNIT WITH UP TO 8 (EIGHT) LINE MODULES.

EACH PASS CONSISTS (FOR 1 LINE) IN :

- 1 LOOP OF 256 SHORT CONTROL FRAMES,
- 1 LOOP OF 256 LONG CONTROL FRAMES,
- 64 LOOPS OF 3 INFORMATION FRAMES.

THERE IS A TOTAL OF 704 TRANSFERS ON EACH LINE  
IN ONE PASS.

2. REQUIREMENTS

HARDWARE: DMS11-F LINE UNIT

3. DMS MODULE PREPARATION

- A. CONNECT ALL THE LINES WITH AN OUTPUT CABLE AND  
A TERMINATOR SOCKET WITH ALL THE SWITCHES ON.
- B. CHECK THAT THE SWITCH ON MODULE M7029 (2P-EM00A-00)  
IS TURNED TOWARDS THE REAR SIDE OF THE RACK  
(NORMAL POSITION).

72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98

4. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:  
DEVADR 160300  
VECTOR 330  
BR1 5  
DEVC 1

5. ERROR REPORT

1ST LINE	KMC REGISTERS (NORMAL ANSWER)
2ND LINE	KMC REGISTERS (MODEM STATUS) !!! OPTIONAL !!!
3RD LINE	- CSR EXTENSION PHYSICAL ADDRESS - CSR EXTENSION EXTENDED ADDRESS (BITS 15-14) - INTERRUPT CALL (# 0 IF NO INTERRUPT) - ADDCON (SEE LISTING) - DATA12 (SEE LISTING)

```

105 000000          IOMODX <DMSA >,160300,330,5,0,0,BUFIN,200,200
      000000          MODULE 150000,DMSA ,160300,330,5,0,0,BUFIN,200,200
                          .TITLE DMSA DEC/X11 SYSTEM EXERCISER  MODULE
                          ;      DDXCOM  VERSION 4(CSSUK)          14-FEB-77
                          ;*****
000000          BEGIN:
000000          104      115      123 MODNAM: .ASCII /DMSA /          ;MODULE NAME.
000003          101      040
000005          000
000006          160300      XFLAG: .BYTE  OPEN          ;TO KEEP TRACK OF WBUFF USAGE
000010          000330      ADDR: 160300+0          ;1ST DEVICE ADDRESS
000012          240      VECTOR: 330+0          ;1ST DEVICE VECTOR
000013          000      BR1: .BYTE  PRTY5+0 ;1ST DEVICE BR LEVEL
000014          000001      BR2: .BYTE  PRTY0+0 ;2ND DEVICE BR LEVEL
000016          000000      DVID1: 0+1          ;DEVICE INDICATOR 1
                          SR1:  OPEN          ;SWITCH REGISTER 1
                          ;*****
000020          150000      STAT: 150000          ;STATUS WORD
000022          000206      INIT:  START          ;MODULE START ADDRESS
000024          000206      SPOINT: MODSP          ;MODULE STACK POINTER
000026          000000      PASCNT: 0          ;PASS COUNTER
000030          000000      ERRCNT: 0          ;ERROR COUNTER
000032          000000      SVR0:  OPEN          ;LOC TO SAVE R0
000034          000000      SVR1:  OPEN          ;LOC TO SAVE R1
000036          000000      SVR2:  OPEN          ;LOC TO SAVE R2
000040          000000      SVR3:  OPEN          ;LOC TO SAVE R3
000042          000000      SVR4:  OPEN          ;LOC TO SAVE R4
000044          000000      SVR5:  OPEN          ;LOC TO SAVE R5
000046          000000      SVR6:  OPEN          ;LOC TO SAVE R6
000050          000000      CSKA:  OPEN          ;ADDRESS OF CURRENT CSR
000052          SBADR:          ;ADDRESS OF GOOD DATA, OR
000052          000000      ACSR:  OPEN          ;CONTENTS OF CSR.
000054          WASADR:          ;ADDRESS OF BAD DATA, OR
000054          000000      ASTAT: OPEN          ;STATUS REGISTER CONTENTS.
000056          000000      ASB:  OPEN          ;EXPECTED DATA.
000060          000000      AWAS:  OPEN          ;ACTUAL DATA.
000062          000250      RSTRT: RESTRT          ;RESTART ADDR AFTER END OF PASS
000064          004134      RBUFVA: BUFIN          ;READ BUFFER VIRTUAL ADDR.
000066          000000      RBUFPA: OPEN          ;READ BUFFER PHYSICAL ADDR.
000070          000000      RBUFEA: OPEN          ;READ BUFFER EA BITS.
000072          000200      RBUFSZ: 200          ;SIZE OF THE READ BUFFER
000074          000000      WBUFPA: OPEN          ;WRITE BUFFER PHYSICAL ADDR.
000076          000000      WBUFEA: OPEN          ;WRITE BUFFER EA BITS
000100          000200      WPUFRO: 200          ;WRITE BUFFER SIZE REQUESTED
000102          000000      WEUFSZ: OPEN          ;WRITE BUFFER SIZE AVAILABLE
000104          000000      FREE:  OPEN          ;RESERVED FOR FUTURE USE
                          .REPT  SPSIZ          ;MODULE STACK STARTS HERE
                          .NLIST
                          .WORD  0
                          .LIST
                          .ENDR

000206          MODSP:
                          ;*****

```

```
107
108
109
110      000000      ECSR=0 ;SPECIALS COMMANDS TO KMC
111      000001      SMAS=1 ;SET MASK
112      000003      DISCO=3 ;DISCONNECT MODEM
113      000004      RXON=4 ;RX ENABLE
114      000005      TXON=5 ;TX ENABLE
115      000006      TXDI=6 ;TX DISABLE
116      000007      RXDI=7 ;RX DISAFLE
117      000010      NORM=10 ;NORMAL MODE
118      000011      MAI1=11 ;MAINTENANCE MODE 1
119      000015      GRUC=15 ;GROUPEd COMMANDS
120
121
122      000000      ; TYPE IN
123      000003      RXIN=0 ;PX INFO FRAME
124      000004      MOST=3 ;MODEM STATUS
125      000005      TXIN=4 ;TX INFO FRAME
126      000006      TXSH=5 ;TX SHOPT CONTROL FRAME
127      000007      TXLO=6 ;TX LONG CONTROL FRAME
127      000007      COMM=7 ;SPECIAL COMMANDS
```



```

158                                     .SBTTL TEST 0
159                                     ;
160                                     ;
161                                     ; DESCRIPTION:
162                                     ; TRANSMISSION AND RECEPTION IN A
163                                     ; LOOPING BACK OF A SHORT CONTROL
164                                     ; FRAME
165                                     ; INITIALIZATION OF THE LINE UNIT
166                                     ; TRANSMISSION OF A RANDOM FRAME
167                                     ; WAITING FOR TWO MESSAGES FROM KMC
168                                     ;     - ACKNOWLEDGE OF THE TRANSMISSION
169                                     ;     - END OF RECEPTION
170                                     ; CHECKING OF THE CONTENT OF THE TWO FRAMES
171                                     ;
172                                     ;
173 000306 016767 003424 003424 DMT00: MOV BAR1,BAR2 ; LINE MASK
174 000314 012767 000001 003420 MOV #1,LIN ; SELECT LINE 0
175 000322 005067 003416 CLR LINUMB ; LINE NUMBER = 0
176                                     ;
177 000326 012767 004400 003456 T00002: MOV #4400,ADDCON ; RETURN VALUE
178 000334 005067 003464 CLR LOOPCN ; CLEAR LOOP COUNT
179 000340 036767 003376 003372 BIT LIN,BAR2 ; IS THE LINE PRESENT ?
180 000346 001520 BEQ T00013 ; NO : NEW LINE
181                                     ;
182 000350 004767 002044 T00003: JSR PC,CLEKMC ; INIT KMC
183                                     ;
184 000354 004767 002220 T00004: JSR PC,RQCSK ; REQUEST CSR REGISTER
185                                     ;
186 000360 T00005:
187 000360 004767 002454 JSR PC,RANDOM
188 000364 116767 002446 003420 MOVE RANNUM,ADDCON ; AVOID 0 IN AD CHARACTER
189 000372 001772 BEQ T00005 ; YES : ANOTHER ONE
190                                     ;
190 000374 016777 003412 003452 MOV ADDCON,@ASEL4 ; LOAD MESSAGE
191 000402 016701 177400 MOV ADDR,R1 ; KMC BASE ADDRESS
192 000406 116761 003332 000003 MOVBL LINUMB,3(R1) ; LINE NUMBER
193 000414 112711 000005 MOVBL #TXSH,(R1) ; SHORT CONTROL
194                                     ;
195 000420 004767 002264 T00006: JSR PC,WTANS ; WAIT ANSWER FROM KMC
196 000424 122777 000025 003420 CMPB #25,@ASEL2 ; END OF TX ?
197 000432 001415 BEQ T00007 ; YES
198 000434 122777 000021 003410 CMPB #21,@ASEL2 ; END OF RX ?
199 000442 001425 BEQ T00010
200                                     ;
201 000444 004767 002414 JSR PC,MESE ; MODEM STATUS
202                                     ;*****
203 000450 104410 000000 004030 ERKNS,BEGIN,TABMOD ;
204                                     ;*****
205 MSGSS,AMES00,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
206 000456 104416 003274 000000 BR T00003 ; TRY AGAIN
207 000464 000731
208                                     ;
209 000466 005777 003360 T00007: IST @ASEL2 ; ERROR FLAG SET ?
210 000472 002352 BGE T00006 ; NO : TRY AGAIN
211                                     ;
212 000474 004767 002364 JSR PC,MESE ; MODEM STATUS
213 000480 000000 004030 ERKNS,BEGIN,TABMOD ;
214 000500 104410 000000 004030
    
```

```

;*****
211 000506 104416 003302 000000 MSGSS,AMES01,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
212 000514 000715 BR T00003 ;TPY AGAIN
213 ;
214 000516 005777 003330 T00010: IST @ASEL2 ; ERROR FLAG SET ?
215 000522 002011 BGE T00011 ; NO : O.K.
216 ;
217 000524 004767 002334 JSR PC,MESE ; MODEM STATUS
218 ;*****
000530 104410 000000 004030 ERRNS,BEGIN,TABMOD ;
;*****
219 000536 104416 003310 000000 MSGSS,AMES02,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
220 000544 000412 BR T00012 ; END
221 ;
222 000546 026777 003240 003300 T00011: CMP ADDCON,@ASEL4 ; COMPARE RECEIVED DATA
223 000554 001406 BEQ T00012 ; O.K.
224 ;*****
000556 104410 000000 004050 ERRNS,BEGIN,TABKMC ;
;*****
225 000564 104416 003316 000000 MSGSS,AMES03,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
226 ;
227 000572 T00012:
000572 104407 000000 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
000576 104407 000000 BREAKS,BEGIN ;CONTINUE AT NEXT INSTRUCTION.
228 000602 105367 003216 DECB LOOPCN ; LOOP COUNT
229 000606 001262 BNE T00004 ; # 0 : LOOP
230 ;
231 000610 005267 003130 T00013: INC LINUMB ; NEW LINE NUMBER
232 000614 000241 CLC
233 000616 006367 003120 ASL LIN ; SHIFT TEST MASK
234 000622 103241 BCC T00002 ; NOT LAST L.U. , LOOP

```

```

236                                     .SBTTL TEST 1
237                                     ;
238                                     ;
239                                     ; DESCRIPTION:
240                                     ; TRANSMISSION AND RECEPTION IN A
241                                     ; LOOPING BACK OF A LONG CONTROL
242                                     ; FRAME
243                                     ; INITIALIZATION OF THE LINE UNIT
244                                     ; TRANSMISSION OF A RANDOM FRAME
245                                     ; WAITING FOR TWO MESSAGES FROM KMC
246                                     ;   - ACKNOWLEDGE OF THE TRANSMISSION
247                                     ;   - END OF RECEPTION
248                                     ; CHECKING OF THE CONTENT OF THE TWO FRAMES
249                                     ;
250                                     ;
251 000624 016767 003106 003106 DMT01: MOV BAR1, BAR2 ; LINE MASK
252 000632 012767 000001 003102 MOV #1, LIN ; SELECT LINE 0
253 000640 005067 003100 CLK LINUMB ; LINE NUMBER = 0
254                                     ;
255 000644 012767 103400 003140 T01002: MOV #103400, ADDCON ; ANSWER
256 000652 005067 003146 CLR LOOPCN ; CLEAR LOOP COUNT
257 000656 036767 003060 003054 BIT LIN, BAR2 ; IS THE LINE PRESENT
258 000664 001550 BEQ T01014 ; NO : NEW LINE
259                                     ;
260 000666 004767 001526 T01003: JSR PC, CLEKMC ; INIT KMC
261                                     ;
262 000672 004767 001702 T01004: JSR PC, RUCSR ; REQUEST CSR REGISTER
263                                     ;
264 000676 T01005:
265 000702 004767 002136 JSR PC, RANDOM
266 000710 116767 002130 003102 MOVB RANNUM, ADDCON ; AVOID 0 IN AD CHARACTER
267 000710 001772 BEQ T01005 ; YES : ANOTHER ONE
268 000712 016777 003074 003134 MOV ADDCON, @ASEL4 ; LOAD IN SEL4
269 000720 004767 002114 JSR PC, RANDOM
270 000724 016777 002106 003126 MOV RANNUM, @ASEL10 ; LOAD IN SEL10
271 000732 016767 002100 003006 MOV RANNUM, DATA12 ; SAVE VALUE
272 000740 004767 002074 JSR PC, RANDOM
273 000744 016777 002066 003110 MOV RANNUM, @ASEL12 ; LOAD IN SEL12
274 000752 016767 002060 002770 MOV RANNUM, DATA3 ; SAVE VALUE
275 000760 016701 177022 MOV ADDR, R1 ; KMC BASE ADDRESS
276 000764 116761 002754 000003 MOVB LINUMB, 3(R1) ; LINE NUMBER
277 000772 112711 000006 MOVB #TXLO, (R1) ; LONG CONTROL
278                                     ;
279 000776 004767 001706 T01006: JSR PC, WTANS ; WAIT ANSWER FROM KMC
280 001002 122777 000026 003042 CMPB #26, @ASEL2 ; END OF TX ?
281 001010 001415 BEQ T01007 ; YES
282 001012 122777 000022 003032 CMPB #22, @ASEL2 ; END OF RX ?
283 001020 001425 BEQ T01010 ; YES
284                                     ;
285 001022 004767 002036 JSR PC, MESE ; MODEM STATUS
286 001026 104410 000000 004030 ;*****
ERRNS, BEGIN, TABMOD ;
;*****
287 001034 104416 003322 000000 MSGSS, AMES04, BEGIN ; ASCII MESSAGE CALL WITH NO HEADER
288 001042 000711 BR T01003 ; TRY AGAIN
289                                     ;
    
```



```

290 001044 005777 003002      T01007: TST      @ASEL2      ; ERROR FLAG SET?
291 001050 002352              HGE      T01006      ; NO : TRY AGAIN
292                               ;
293 001052 004767 002006      JSR      PC,MESE      ; MODEM STATUS
294                               ;*****
001056 104410 000000 004030  ERKNS,BEGIN,TABMOD      ;
                               ;*****
295 001064 104416 003330 000000  MSGSS,AMES05,BEGIN      ;ASCII MESSAGE CALL WITH NO HEADER
296 001072 000675              BR      T01003      ; TRY AGAIN
297                               ;
298 001074 005777 002752      T01010: TST      @ASEL2      ; ERROR FLAG SET ?
299 001100 002011              HGE      T01011      ; NO : O.K.
300                               ;
301 001102 004767 001756      JSR      PC,MESE      ; MODEM STATUS
302                               ;*****
001106 104410 000000 004030  ERRNS,BEGIN,TABMOD      ;
                               ;*****
303 001114 104416 003336 000000  MSGSS,AMES06,BEGIN      ;ASCII MESSAGE CALL WITH NO HEADER
304 001122 000422              BR      T01013      ; END
305                               ;
306 001124 026777 002662 002722 T01011: CMP      ADDCON,@ASEL4      ; FRAME ?
307 001132 001010              BNE      T01012      ; BAD
308 001134 026777 002606 002716  CMP      DATA2,@ASEL10 ; DATA 1,2 ?
309 001142 001004              BNE      T01012      ; BAD
310 001144 126777 002600 002710  CMPB     DATA3,@ASEL12 ; DATA 3 ?
311 001152 001406              BEQ      T01013      ; GOOD : END
312                               ;
313 001154              T01012:
                               ;*****
001154 104410 000000 004050  ERRNS,BEGIN,TABKMC      ;
                               ;*****
314 001162 104416 003344 000000  MSGSS,AMES07,BEGIN      ;ASCII MESSAGE CALL WITH NO HEADER
315                               ;
316 001170              T01013:
001170 104407 000000  BREAKS,BEGIN      ;TEMPORARY RETURN TO MONITOR...
001174 104407 000000  BPEAKS,BEGIN      ;CONTINUE AT NEXT INSTRUCTION.
317 001200 105367 002620      DECB     LOOPCN      ; LOOP COUNT
318 001204 001232              BNE      T01004      ; # 0 : LOOP
319                               ;
320 001206 005267 002532      T01014: INC      LINUMB      ; NEW LINE NUMBER
321 001212 000241              CLC
322 001214 006367 002522      ASL      LIN      ; SHIFT TEST MASK
323 001220 103211              BCC      T01002      ; NOT LAST L.U. , LOOP
    
```

```

325 .SBTTL TEST 2
326 ;
327 ;
328 ;
329 ;
330 ;
331 ;
332 ;
333 ;
334 ;
335 ;
336 ;
337 ;
338 ;
339 ;
340 ;
341 ;
342 ;
343 001222 012767 000100 002574 DMT02: MOV #100,LOOPCN ; INIT LOOP COUNT
344 ;
345 001230 T02002:
    001230 104413 003762 GETPA$,INFO2 ;GET PHYSICAL ADDRESS OF INFO2
346 001234 006367 002526 ASL INFO2+4 ; ROTATE EXTENSION BITS
347 001240 006367 002522 ASL INFO2+4
348 001244 000367 002516 SWAB INFO2+4 ; TO BITS 15-14
349 001250 112767 000377 002510 MOVEB #377,INFO2+4 ; MAXIMUM BIT COUNT
350 001256 016767 002454 002454 MOV BAR1,BAR2 ; LINE MASK
351 001264 012767 000001 002450 MOV #1,LIN ; SELECT LINE 0
352 001272 005067 002446 CLR LINUMB ; LINE NUMBER = 0
353 ;
354 001276 012767 000003 002444 T02003: MOV #3,DATA3 ; 3 TRANSMISSIONS BY L.U.
355 001304 036767 002432 002426 BIT LIN,BAR2 ; IS THE LINE PRESENT ?
356 001312 001002 BNE T02004 ; YES : TRY IT
357 001314 000167 000522 JMP T02024 ; NO : JUMP
358 ;
359 001320 004767 001074 T02004: JSR PC,CLEKMC ; INIT KMC
360 ;
361 001324 004767 001250 T02005: JSR PC,ROCSR ; REQUEST REGISTERS
362 001330 104413 003770 GETPA$,ABUFIN ;GET PHYSICAL ADDRESS OF ABUFIN
363 001334 016777 002432 002516 MOV ABUFIN+2,@ASEL10
364 001342 006367 002426 ASL ABUFIN+4 ; ROTATE EXTENSION BITS
365 001346 006367 002422 ASL ABUFIN+4
366 001352 000367 002416 SWAB ABUFIN+4 ; TO BITS 15-14
367 001356 016777 002412 002476 MOV ABUFIN+4,@ASEL12
368 001364 005077 002464 CLR @ASEL4 ; CLEAR SEL4
369 001370 016761 176412 MOV ADDR,R1 ; KMC BASE ADDRESS
370 001374 116761 002344 000003 MOVEB LINUMB,3(R1) ; LINE NUMBER
371 001402 112711 000000 MOVEB #RXIN,(R1) ; RX IN
372 001406 105077 002440 CLRR @ASEL2 ; CLEAR BSEL2
373 001412 105327 1s: DECB (PC)+ ; TEMPO
374 001414 000000 0
375 001416 001375 BNE 1s
376 001420 004767 001154 JSR PC,ROCSR ; REQUEST REGISTERS
377 001424 104412 000000 GWBUFS,HEGIN ;GET WRITE BUFFER INFORMATION
378 001430 016767 176440 002314 MOV WBUFEA,INFO1 ; WRITE BUFFER ADDRESS
379 001436 116767 176434 002311 MOVEB WBUFEA,INFO1+3
380 001444 106367 002305 ASLR INFO1+3 ; ROTATE EXTENSION BITS
  
```



```
430
431 001732 026777 002054 002114 ;
432 001740 001004 T02016: CMP ADDCON,@ASEL4 ; FRAME ?
433 001742 126777 002000 002110 BNE T02017 ; BAD
434 001750 001407 CMPB DATA12,@ASEL10 ; CHARACTER COUNT ?
435 BFG T02020 ; GOOD
436 001752 ;
;*****
ERRNS,BEGIN,TABKMC ;
;*****
437 001760 104416 003372 000000 MSGSS,AMES11,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
438 001766 000425 BR T02024 ; END
439 ;
440 001770 T02020: GETPAS,RBUFVA ;GET PHYSICAL ADDRESS OF RBUFVA
001770 104413 000064 MOVB DATA12,RBUFSZ
441 001774 116767 001746 176070 ASH RBUFSZ ; BUFFER SIZE IN WORDS
442 002002 006267 176064 BEQ T02022 ; = 0 : END
443 002006 001404 CDATAS,BEGIN,RBUFP A ;REQUEST MONITOR TO CHECK DATA
444 002010 104414 000000 000066 .+2 ;IF ERROR, CONTINUE
002016 002020
445 ;
446 002020 T02022: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
002020 104407 000000 BREAKS,BEGIN ;CONTINUE AT NEXT INSTRUCTION.
002024 104407 000000 DEC DATA3 ; ONE MORE TIME FOR THIS L.U. ?
447 002030 005367 001714 BEQ T02024 ; NO : CHANGE L.U.
448 002034 001402 JMP T02005 ; YES : LOOP
449 002036 000167 177262 ;
450 ;
451 002042 005267 001676 T02024: INC LINUMB ; NEW LINE NUMBER
452 002046 000241 CLC
453 002050 006367 001666 ASL LIN ; SHIFT TEST MASK
454 002054 103402 BCS T02030 ; THE END
455 002056 000167 177214 JMP T02003 ; NOT LAST L.U. , LOOP
456 ;
457 002062 005367 001736 T02030: DEC LOOPCN ; LOOP COUNT
458 002066 001402 BEQ PASS ; = 0 : END OF PASS
459 002070 000167 177134 JMP T02002 ; # 0 : LOOP
```

461  
462  
463  
464

465 002074  
002074 104402 000000'

;  
;  
;  
PASS:

.SBTTL END OF PASS

ENDPASS,BEGIN ;SIGNAL END OF PASS, CONTINUE AT RESTRT

```

467                                     .SBTTL  AUTOMATIC LINE CONFIGURATION
468                                     ;
469                                     ;
470                                     ;
471                                     ;
472                                     ;
473                                     ;
474                                     ;
475                                     ;
476                                     ;
477                                     ;
478                                     ;
479                                     ;
480                                     ;
481                                     ;
482                                     ;
483                                     ;
484                                     ;
485 002100 016701 175702 LINAUT: MOV ADDR,R1 ; INIT CSR
486 002104 012767 177777 001624 MOV #-1,BAR1 ; INIT PARAMETERS
487 002112 012767 177777 001620 MOV #-1,BAR2
488 002120 012767 000001 001614 MOV #1,LIN
489 002126 005067 001612 CLR LINUMB
490                                     ;
491 002132 005011 3s: CLR (R1) ; CLEAR SELO
492 002134 012711 000100 MOV #100,(R1) ; MASTER CLEAR
493 002140 005011 CLR (R1) ; CLEAR SELO
494 002142 012767 122010 001560 MOV #244*400+10,INSTLO
495 002150 016761 001554 000006 MOV INSTLO,6(R1) ; ADRESS LINE
496 002156 112761 000002 000001 MOVB #2,1(R1)
497 002164 116711 001554 MOVB LINUMB,(R1) ; LINE NUMBER
498 002170 112761 000003 000001 MOVB #3,1(R1)
499 002176 105061 000001 CLRB 1(R1) ; CHOOSE A LINE NUMBER
500                                     ;
501 002202 012767 021200 001520 MOV #42*400+200,INSTLO
502 002210 016761 001514 000006 MOV INSTLO,6(R1) ; LINE TEST
503 002216 112761 000002 000001 MOVB #2,1(R1)
504 002224 112761 000003 000001 MOVB #3,1(R1)
505 002232 105061 000001 CLRB 1(R1) ; READ IF ON LINE IN (R1)
506                                     ;
507 002236 111167 001472 MOVB (R1),BAD
508 002242 012767 000020 001462 MOV #20,GOOD
509 002250 056767 001470 001454 BIS LINUMB,GOOD ; COMPARE WITH EXPECTED VALUE
510 002256 126767 001452 001446 CMPB BAD,GOOD
511 002264 001403 BEQ 7S
512                                     ;
513 002266 046767 001450 001442 BIC LIN,BAR1 ; NOT HERE, SUPPRESS IT
514                                     ;
515 002274 005267 001444 7s: INC LINUMB
516 002300 000241 CLC
517 002302 006367 001434 ASL LIN
518 002306 103311 BCC 3S
519 002310 000207 RTS PC

```

```

521                                .SHTTL KMC CODE LOADER
522                                ;
523                                ;
524                                ; DESCRIPTION:
525                                ;
526                                ; ROUTINE TO LOAD KMC CODE
527                                ;
528                                ; CALLING SEQUENCE:
529                                ;
530                                ; JSR PC,LOADER
531                                ;
532                                ;
533                                ;
534 002312 016700 006216          LOADER: MOV KMCODE,R0          ; LOAD CODE IN KMC CRAM
535 002316 016701 175464          MOV ADDR,R1             ; BASE ADDRESS
536 002322 012702 001777          MOV #1777,R2
537                                ;
538 002326 105061 000001          1S: CLRR 1(R1)
539 002332 112761 000004 000001  MOVB #4,1(R1)          ; SET RAM0 TO ONE
540 002340 010261 000004          MOV R2,4(R1)          ; LOAD CRAM ADDRESS
541 002344 014061 000006          MOV -(R0),6(R1)       ; LOAD MICRO-INSTRUCTION
542 002350 152761 000040 000001  BISR #40,1(R1)       ; CRAM WRITE
543 002356 142761 000040 000001  BICB #40,1(R1)       ; CLEAR CRAM WRITE
544 002364 021061 000004          CMP (R0),4(R1)       ; CHECK INSTRUCTION
545 002370 001003          BNE 2S
546                                ;
547 002372 005302          DEC R2
548 002374 002354          BGE 1S
549                                ;
550 002376 000207          RTS PC
551                                ;
552 002400          2S:
                    002400 104410 000000 004050  ;*****
ERRNS,BEGIN,TABKMC ;
                    ;*****
553 002406 104416 003400 000000  MSGSS,AMES12,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
554 002414 104403 000000  ENDS,BEGIN ; BAD KMC LOAD , DROP THE MODULE

```

```

556 .SBTTL CLEAR KMC
557 ;
558 ;
559 ; DESCRIPTION:
560 ;
561 ; CLEAR CSR REGISTERS TOWARD KMC
562 ;
563 ; CALLING SEQUENCE:
564 ;
565 ; JSR PC,CLEKMC
566 ;
567 ;
568 ;
569 002420 016701 175362 CLEKMC: MOV ADDR,R1 ; BASE ADDRESS
570 002424 010100 MOV R1,R0
571 002426 005020 CLR (R0)+
572 002430 112761 000100 000001 MOVB #100,1(R1) ; MCLR
573 002436 005020 CLR (R0)+ ; INIT CSR
574 002440 005020 CLR (R0)+
575 002442 005010 CLR (R0)
576 002444 112761 000200 000001 MOVB #200,1(R1) ; START KMC CODE
577 002452 004767 000122 JSR PC,RQCSR ; REQUEST CSR REG
578 002456 104413 004004 GETPAS,ACSREX ; GET PHYSICAL ADDRESS OF ACSREX
579 002462 016777 001320 001364 MOV ACSREX+2,@ASEL4 ; PHYSICAL ADDRESS IN KMC
580 002470 006367 001314 ASL ACSREX+4 ; ROTATE EXTENSION ADDRESS
581 002474 006367 001310 ASL ACSREX+4
582 002500 000367 001304 SWAB ACSREX+4 ; TO BITS 15-14
583 002504 016777 001300 001344 MOV ACSREX+4,@ASEL6 ; IN KMC
584 002512 112777 000007 001330 MOVB #COMM,@ASEL0
585 002520 004767 000164 JSR PC,WIANS ; WAIT ACKNOWLEDGE
586 002524 005777 001322 TST @ASEL2 ; ERROR FLAG SET ?
587 002530 002404 BLT 1S ; YES : ERROR
588 ;
589 002532 122777 000027 001312 CMPB #27,@ASEL2 ; GOOD ANSWER ?
590 002540 001406 BEQ 2S ; YES : O.K.
591 ;
592 002542 1S:
;*****
ERRNS,BEGIN,TABKMC ;
;*****
MSGSS,AMES15,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
593 002550 104416 003420 000000 MSGSS,AMES15,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
594 ;
595 002556 012700 000001 2S: MOV #SMAS,R0 ; SET MASK TO ZERO
596 002562 004767 000406 JSR PC,GROU ; EXECUTE
597 002566 012700 000015 MOV #GROC,R0 ; SET INIT GROUP COMMAND
598 002572 004767 000376 JSR PC,GROU ; EXECUTE
599 002576 000207 RTS PC

```



```

601                                     .SBTTL REQUEST CSR REGISTERS (RDYI INTERRUPT)
602                                     ;
603                                     ;
604                                     ; DESCRIPTION:
605                                     ;
606                                     ; REQUEST CSR REGISTERS
607                                     ; LOAD RQI IN CSR 0
608                                     ; WAIT FOR RDYI INTERRUPT (CSR 2)
609                                     ;
610                                     ; CALLING SEQUENCE:
611                                     ;
612                                     ; JSR PC,RQCSR
613                                     ;
614 002600 005067 001210 RQCSR: CLR CLK ; CLEAR LOOP
615 002604 005067 001114 CLR INTFLG ; INIT THE SOFTWARE INTERRUPT FLAG
616 002610 005067 001212 CLR CALLFC ; CLEAR CALL ADDRESS
617 002614 105077 001232 CLKB @ASEL2 ; CLEAR BSEL2
618 002620 112777 000300 001222 MOVV #300,@ASEL0 ; SET RQI & IEI
619                                     ;
620 002626 RQCSR1:
621 002626 104407 000000 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
622 002632 104407 000000 BREAKS,BEGIN ;CONTINUE AT NEXT INSTRUCTION.
623 002636 005767 001062 TST INTFLG ; DID THE INTERRUPT OCCURS ?
624 002642 001013 BNE RQCSR3 ; YES : NORMAL RETURN
625                                     ;
626 002644 005367 001144 DEC CLK ; DECREMENT TIMING
627 002650 001366 BNE RQCSR1 ; NO TIME OUT : LOOP
628                                     ;
629 002652 011667 001150 MOV (SP),CALLPC ; SET CALL ADDRESS
630 002656 104416 003404 000000 MSGSS,AMES13,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
631 002664 042777 000300 001156 BIC #300,@ASEL0 ; CLEAR RQI & IEI
632                                     ;
633 RQCSR3: RTS PC ; NORMAL RETURN
634                                     ;
635                                     ;
636 002674 042777 000300 001146 INTRDI: BIC #300,@ASEL0 ; CLEAR RQI & IEI
637 002702 005167 001016 COM INTFLG ; SET THE INTERRUPT FLAG
638 002706 000002 RTI ; RETURN TO CONTROL

```

```

640          .SETTL WAIT KMC ANSWER (RDYO INTERRUPT)
641          ;
642          ;
643          ; DESCRIPTION :
644          ;
645          ; REQUEST KMC FUNCTION
646          ; WAIT FOR RDYO INTERRUPT (CSR 2)
647          ;
648          ; CALLING SEQUENCE :
649          ;
650          ; JSR      PC,WTANS
651          ;
652          ;
653          ;
654 002710 012767 000003 001100 WTANS: MOV      #3,CLK1      ; INIT RETRY COUNT
655 002716 005067 001002          CLR      INTFLG      ; INIT THE SOFTWARE INTERRUPT FLAG
656 002722 005067 001100          CLR      CALLPC      ; CLEAR CALL ADDRESS
657          ;
658 002726 005067 001062          WTANS1: CLR      CLK          ; CLEAR LOOP
659 002732 105077 001114          CLR      @ASEL2      ; CLEAR BSEL2
660 002736 052777 000020 001104          BIS      #20,@ASEL0 ; SET IEO
661          ;
662 002744          WTANS2:
663 002744 104407 000000          BREAKS,BEGIN      ; TEMPORARY RETURN TO MONITOR...
664 002750 104407 000000          BREAKS,BEGIN      ; CONTINUE AT NEXT INSTRUCTION.
665          ;
666 002754 005767 000744          IST      INTFLG      ; DID INTERRUPT OCCURS ?
667 002760 001016          BNE      WTANS3      ; YES : NORMAL RETURN
668          ;
669 002762 005367 001026          DEC      CLK          ; DECREMENT TIMING
670 002766 001366          BNE      WTANS2      ; NO TIME OUT : LOOP
671          ;
672 002770 005367 001022          DFC      CLK1         ; DECREMENT COUNT
673 002774 001354          BNE      WTANS1      ; # 0 : RETRY
674          ;
675 002776 011667 001024          MOV      (SP),CALLPC ; SET CALL ADDRESS
676 003002 104416 003412 000000 MSGSS,AMES14,BEGIN ; ASCII MESSAGE CALL WITH NO HEADER
677 003010 042777 000020 001032          BIC      #20,@ASEL0 ; CLEAR IEO
678          ;
679          ;
680          ;
681 003020 042777 000020 001022          INTRDO: BIC      #20,@ASEL0 ; CLEAR IEO
682 003026 005167 000672          COM      INTFLG      ; SET THE INTERRUPT FLAG
683 003032 000002          RTI          ; RETURN TO CONTROL
    
```

```
685  
686  
687  
688  
689 003034 000000  
690 003036 000000  
691  
692  
693  
694  
695 003040 017767 177770 177770  
696 003046 062767 000002 177760  
697 003054 100002  
698  
699 003056 005067 177752  
700  
701 003062 000207  
; .SBTTL RANDOM NUMBER GENERATOR SUBROUTINE  
;  
;  
RANNAD: 0 ; RANDOM ADDRESS NUMBER  
RANNUM: 0 ; RANDOM NUMBER VALUE  
;  
;  
RANDOM: MOV @RANNAD,RANNUM ; RANDOM VALUE  
ADD #2,RANNAD ; INCREASE RANDOM ADDRESS  
BPL 1$  
;  
CLR RANNAD ; CLEAR RANDOM ADDRESS IF > 100000  
1$: RTS PC
```

```

703          .SBTTL  ERROR MESSAGE FROM KMC
704          ;
705          ;
706          ; DESCRIPTION :
707          ;
708          ;       SAVE KMC STATUS
709          ;       REQUEST MODEM STATUS
710          ;       WAIT FOR KMC ANSWER
711          ;
712          ; CALLING SEQUENCE :
713          ;
714          ;       JSR      PC,MESE
715          ;
716          ;
717          ;
718          ;
719 003064 017767 000760 001022 MESE:  MOV      @ASEL0,BUFKMC      ; SAVE SEL0
720 003072 017767 000754 001016      MOV      @ASEL2,BUFKMC+2    ; SAVE SEL2
721 003100 017767 000750 001012      MOV      @ASEL4,BUFKMC+4    ; SAVE SEL4
722 003106 017767 000744 001006      MOV      @ASEL6,BUFKMC+6    ; SAVE SEL6
723 003114 017767 000740 001002      MOV      @ASEL10,BUFKMC+10  ; SAVE SEL10
724 003122 017767 000734 000776      MOV      @ASEL12,BUFKMC+12  ; SAVE SEL12
725 003130 017767 000730 000772      MOV      @ASEL14,BUFKMC+14  ; SAVE SEL14
726 003136 017767 000724 000766      MOV      @ASEL16,BUFKMC+16  ; SAVE SEL16
727 003144 004767 177430      JSR      PC,RQCSR          ; REQUEST CSR REGISTER
728 003150 016701 174632      MOV      ADDR,R1          ; KMC BASE ADDRESS
729 003154 116761 000564 000003      MOVB    LINUMB,3(R1)      ; LINE NUMBER
730 003162 112711 000003      MOVB    #MUST,(R1)
731 003166 004767 177516      JSR      PC,WTANS          ; WAIT RDYO
732 003172 000207
RTS      PC
  
```



```
770 .SBTTL ASCII STRINGS
771 ;
772 ;
773 ;
774 003274 003430* 003455* 177777 AMES00: MESS00,MESS03,177777
775 003302 003430* 003501* 177777 AMES01: MESS00,MESS04,177777
776 003310 003430* 003515* 177777 AMES02: MESS00,MESS05,177777
777 003316 003531* 177777 AMES03: MESS06,177777
778 ;
779 003322 003437* 003455* 177777 AMES04: MESS01,MESS03,177777
780 003330 003437* 003501* 177777 AMES05: MESS01,MESS04,177777
781 003336 003437* 003515* 177777 AMES06: MESS01,MESS05,177777
782 003344 003550* 177777 AMES07: MESS07,177777
783 ;
784 003350 003446* 003455* 177777 AMES08: MESS02,MESS03,177777
785 003356 003446* 003501* 177777 AMES09: MESS02,MESS04,177777
786 003364 003446* 003515* 177777 AMES10: MESS02,MESS05,177777
787 003372 003446* 003566* 177777 AMES11: MESS02,MESS08,177777
788 ;
789 003400 003603* 177777 AMES12: MESS09,177777
790 ;
791 003404 003630* 003650* 177777 AMES13: MESS10,MESS12,177777
792 ;
793 003412 003640* 003650* 177777 AMES14: MESS11,MESS12,177777
794 ;
795 003420 003664* 177777 AMES15: MESS13,177777
796 ;
797 003424 003703* 177777 AMES16: MESS14,177777
```

799	003430	123	056	103	MESS00: .ASCIZ /S.C. /
	003433	056	040	040	
	003436	000			
800	003437	114	056	103	MESS01: .ASCIZ /L.C. /
	003442	056	040	040	
	003445	000			
801	003446	106	122	101	MESS02: .ASCIZ /FRAME /
	003451	115	105	040	
	003454	000			
802	003455	125	116	105	MESS03: .ASCIZ /UNEXPECTED MESSAGE%/
	003460	130	120	105	
	003463	103	124	105	
	003466	104	040	115	
	003471	105	123	123	
	003474	101	107	105	
	003477	045	000		
803	003501	102	101	104	MESS04: .ASCIZ /BAD TX END%/
	003504	040	124	130	
	003507	040	105	116	
	003512	104	045	000	
804	003515	102	101	104	MESS05: .ASCIZ /BAD RX END%/
	003520	040	122	130	
	003523	040	105	116	
	003526	104	045	000	
805	003531	123	110	117	MESS06: .ASCIZ /SHORT CONTROL%/
	003534	122	124	040	
	003537	103	117	116	
	003542	124	122	117	
	003545	114	045	000	
806	003550	114	117	116	MESS07: .ASCIZ /LONG CONTROL%/
	003553	107	040	103	
	003556	117	116	124	
	003561	122	117	114	
	003564	045	000		
807	003566	102	125	106	MESS08: .ASCIZ /BUFFER SIZE%/
	003571	106	105	122	
	003574	040	123	111	
	003577	132	105	045	
	003602	000			
808	003603	113	115	103	MESS09: .ASCIZ /KMC CODE LOAD ERROR%/
	003606	040	103	117	
	003611	104	105	040	
	003614	114	117	101	
	003617	104	040	105	
	003622	122	122	117	
	003625	122	045	000	
809	003630	116	117	040	MESS10: .ASCIZ /NO RDY1/
	003633	122	104	131	
	003636	111	000		
810	003640	116	117	040	MESS11: .ASCIZ /NO RDY0/
	003643	122	104	131	
	003646	117	000		
811	003650	040	111	116	MESS12: .ASCIZ / INTERRUPT%/
	003653	124	105	122	
	003656	122	125	120	
	003661	124	045	000	
812	003664	103	123	122	MESS13: .ASCIZ /CSR EXTENSION%/

	003667	040	105	130	
	003672	124	105	116	
	003675	123	111	117	
	003700	116	045	000	
813	003703	123	120	105	MESS14: .ASCIZ /SPECIAL COMMAND%/
	003706	103	111	101	
	003711	114	040	103	
	003714	117	115	115	
	003717	101	116	104	
	003722	045	000		
814					.EVEN



816  
817  
818  
819  
820 003724 000000  
821 003726 000000  
822 003730 000  
823 003731 000  
824 003732 000000  
825 003734 000000  
826 003736 000000  
827 003740 000000  
828 003742 000000  
829 003744 000000  
830 003746 000000  
831 003750 000000  
832 003752 000000  
833 003754 000000  
834 003756 000000  
835 003760 000000  
836 003762 004134  
837 003764 000000  
838 003766 000000  
839 003770 003764  
840 003772 000000  
841 003774 000000  
842 003776 003752  
843 004000 000000  
844 004002 000000  
845 004004 004104  
846 004006 000000  
847 004010 000000  
848  
849 004012 000000  
850 004014 000000  
851 004016 000000  
852 004020 000000  
853 004022 000000  
854 004024 000000  
855 004026 000000

.SBTTL PROGRAM VARIABLES  
;  
;  
;  
INTFLG: 0  
MASK: 0  
INSTLO: .BYTE 0  
INSTHI: .BYTE 0  
GOOD: 0  
BAD: 0  
BAK1: 0  
BAK2: 0  
LIN: 0  
LINUMB: 0  
DATA12: 0  
DATA3: 0  
INFO1: 0  
0  
0  
0  
INFO2: BUFIN  
0  
0  
ABUFIN: INFO2+2  
0  
0  
ARUFOU: INFO1  
0  
0  
0  
ACSREX: CSREX  
0  
0  
;  
ADDCON: 0  
CLK: 0  
CLK1: 0  
BYCO: 0  
TYPCOM: 0  
LOOPCN: 0  
CALLPC: 0

```

      857                                .SHTTL  BUFFERS
      858                                ;
      859                                ;
      860                                ;
      861 004030 004114'                TABMOD:  BUFKMC
      862 004032 004116'                BUFKMC+2
      863 004034 004120'                BUFKMC+4
      864 004036 004122'                BUFKMC+6
      865 004040 004124'                BUFKMC+10
      866 004042 004126'                BUFKMC+12
      867 004044 004130'                BUFKMC+14
      868 004046 004132'                BUFKMC+16
      869 004050                                TABKMC:
      870 004050 000000                ASEL0:  0
      871 004052 000000                ASEL2:  0
      872 004054 000000                ASEL4:  0
      873 004056 000000                ASEL6:  0
      874 004060 004104'                ASEL10: CSREX
      875 004062 004106'                ASEL12: CSREX+2
      876 004064 004110'                ASEL14: CSREX+4
      877 004066 004112'                ASEL16: CSREX+6
      878 004070 004006'                ACSREX+2
      879 004072 004010'                ACSREX+4
      880 004074 004026'                CALLPC
      881 004076 004012'                ADDCOM
      882 004100 003746'                DATA12
      883 004102 177777                177777
      884                                ;
      885                                ;
      886                                ;
      887 004104                CSREX:  .BLKW  4
      888 004114                BUFKMC:  .BLKW  10
      889 004134                BUFIN:   .BLKB  400
```

2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58

004534

.SBTTL \*KMC\* \*\*\*\* DMS11F KMC MICROCODE \*\*\*\*

.LIST MC  
.NLIST ME

KSTART:

; .IDENT /X01.01/  
;  
; \*\*\*\*\*  
; \* DMS11F \*  
; \*\*\*\*\*

COPYRIGHT (C) 1978 BY  
DIGITAL EQUIPMENT FRANCE, ANNECY

; THIS SOFTWARE IS FURNISHED UNDER A LICENCE AND MAY BE USED AND COPIED  
; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH A LICENCE AND WITH THE  
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
; OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
; TRANSFERRED.

; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT.

; DIGITAL ACCEPTS NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

; PROGRAMMER/DATE: MICHEL BLANC 20/3/78

; ABSTRACT:

; FIRMWARE FOR DMS11-F PROJECT  
; KMC11 PROGRAM FOR  
; MULTIPLEXER  
; -UP TO 8 LINE UNITS  
; -SPEED : EACH LINE AT ANY SPEED  
; UP TO 72000 BAUDS  
; -TRANSMISSION AND RECEPTION MAY  
;  
; BE AT DIFFERENT SPEEDS  
; -FULL DUPLEX  
; -DMA TRANSFER FOR EACH LINE  
; WITH POSSIBILITY TO SPLIT  
; MEMORY IN SEVERAL BLOCKS  
; -ERROR RECOGNITION AND HANDLING  
; -RECOGNITION OF CONTROL FRAME

\*KMC\* \*\*\*\* DMS11F KMC MICROCODE \*\*\*\*

```
59 ; DATE : APRIL 14, 1978
60 ;
61 ;
62 ; VERSION: 03
63 ;
64 ; HISTORY:
65 ; 14 APRIL 1978 : CREATION
66 ; 23 JULY 1979 : ADD ROUTINE FOR GARBAGE RECEIVED DATA WHEN
67 ; NO BUFFER IS AVAILABLE
68 ; 30 OCTOBER 1979 : SUPPRESS ROUTINES FOR ERROR CLASSES
69 ; MODIFY RECEPTION BUFFER HANDSHAKING ( NO
70 ; MORE SUPPRESSION WHEN ERRONEOUS CONTROL
71 ; FRAME RECEIVED )
72 ; MODIFY SCAN REGISTER HANDSHAKING
```

```

74          .SBTTL *KMC*  DEFINITIONS
75          ;
76          ; *****
77          ; * ASSEMBLER PARAMETERS *
78          ; *****
79          ;
80          ;
81          ;
82          ;
83          ; ODDADD : FOR BREAKPOINT WHEN FIRST CHARACTER IS > 3
84          ; TXER   : FOR TX ERROR
85          ; BKP    : OTHER BREAKPOINTS
86          ;
87          ; BRG = 0 : S O M UNEXPECTED DURING FRAME
88          ; BRG = 1 : NO S O M AT BEGINNING OF FRAME
89          ; BRG = 2 : TX ERROR
90          ; BRG = 4 : ADDRESS CHARACTER > 3
91          ; BRG = 17: RECEPTION BUFFER IS TOO SMALL
92          ;
93          ;
94          ;ODDADD = 0
95          ;BKP = 0
96          ;TXER = 0
97          ;
98          ;
99          ;
100         ;
101         ; *****
102         ; * DEFINITIONS *
103         ; *****
104         ;
105         ;
106         ; MACRO RE-DEFINITION OF LDMAP
107         ;
108         ;
109         .MACRO LDMAP SRC,DATA
110             <MOVE!LDMAPG!IMM!<DATA>>
111         .ENDM
112         ;
113         ;
114         ; MASK FOR SCAN STATUS
115         ;
116         000277 MFII=277          ; FIFO INPUT READY
117         000337 MFIOU=337        ; FIFO OUTPUT FREE
118         000357 MSTIN=357        ; STATUS INPUT LOADED
119         000367 MSTOU=367        ; STATUS OUTPUT READ
120         ;
121         ;
122         ; LINE UNIT REGISTERS DEFINITION
123         ;
124         ; INPUT
125         ;
126         000200 RELUAD=200        ; DECODED L.U. ADDRESS
127         000240 RESCAN=240       ; SCAN STATUS
128         000300 REFICO=300       ; FIFO CONTROL
129         000320 REFIDA=320       ; FIFO DATA
130         000340 RESTLO=340       ; STATUS LOW
    
```

## \*KMC\* DEFINITIONS

```

131          000360          RESTH1=360          ; STATUS HIGH
132          ;
133          ; OUTPUT
134          ;
135          002010          LολουAD=10!2000          ; L.U. ADDRESS
136          002012          STSCAN=12!2000          ; START SCAN
137          002014          LOFICU=14!2000          ; FIFO CONTROL
138          002015          LOFIDA=15!2000          ; FIFO DATA
139          002016          LOSTLO=16!2000          ; STATUS LOW
140          002017          LOSTHI=17!2000          ; STATUS HIGH
141          ;
142          ;
143          ; SCRATCH PAD REDEFINITIONS
144          ;
145          000010          SPR=10
146          000011          SP9=11
147          000012          SP10=12
148          000013          SP11=13
149          000014          SP12=14
150          000015          SP13=15
151          000016          SP14=16
152          000017          SP15=17
153          ;
154          ;
155          ; ERRORS DETECTED BY KMC11
156          ;
157          000100          ABORT=100          ; ABORT RECEPTION RQ BY PDP
158          000001          NOBUFF=1          ; NO BUFFER AVAILABLE
159          000002          BUFOVE=2          ; RECEIVER BUFFER OVFL
160          000004          NOEXME=4          ; NON-EXIST. MEMORY
161          000010          NOIPRE=10          ; L.U. NOT PRESENT
162          000020          BANOLO=20          ; ABORT TRANSMISSION RQ BY PDP
163          000040          INVCOM=40          ; INVALID COMMAND
164          000200          LUBUSY=200          ; L.U. BUSY
165          ;
166          000040          FCSERR=40          ; INVALID LENGTH
167          ;
168          ; TRANSMISSION STATES
169          ;
170          000000          KFREE=0          ; FREE
171          000002          BEGFRA=2          ; BEGIN OF FRAME
172          000003          BEBUOD=3          ; BEGIN OF BUFFER ODD
173          000004          SECHAR=4          ; SECOND CHARACTER
174          000005          FICHAR=5          ; FIRST CHARACTER
175          000006          SFADDR=6          ; SHORT FRAME : ADDRESS
176          000007          SFCTRL=7          ; SHORT FRAME : CONTROL
177          000014          LFADDR=14          ; LONG FR: ADDRESS
178          000013          LFCTRL=13          ; LONG FR: CONTROL
179          000012          LFDAT1=12          ; LONG FR: DATA 1
180          000011          LFDAT2=11          ; LONG FR: DATA 2
181          000010          LFDAT3=10          ; LONG FR: DATA 3
182          000015          TXABOR=15          ; TRANSMISSION ABORT
183          000016          TXNOEM=16          ; NON EXISTENT MEMORY
184          ;
185          ; RECEPTION STATES
186          ;
187          000000          RXFREE=0          ; RXFREE

```

188	000001	BFEF=1	; BEGIN FRAME + EOF
189	000002	WTCIRL=2	; WT CONTROL CHARACTER
190	000003	CTRL=3	; CONTROL CHAR + EOF
191	000004	LFD1=4	; LONG FRAME + DATA1
192	000005	LFD1EF=5	; IDEM + EOF
193	000006	LFD2=6	; LONG FRAME + DATA2
194	000007	LFD2EF=7	; IDEM + EOF
195	000010	LFD3=10	; LONG FRAME + DATA3
196	000011	LFD3EF=11	; IDEM + EOF
197	000012	ABOF=12	; ABORT PECEPTION
198	000013	ABOPEF=13	; UDEM + EOF
199	000014	ABORR=14	; ABORT REQUESTED BY PDP
200	000015	ABORRF=15	; IDEM + EOF
201	000016	INOR=16	; INFO FRAME, NO BUFFER
202	000017	INOBEF=17	; IDEM + EOF
203	000020	IN1E=20	; INFO, NEW BUF, CHAR 1, EVEN
204	000021	IN1EEF=21	; IDEM = EOF
205	000022	IN1O=22	; INFO, NEW BUF, CHAR 1, ODD
206	000023	IN1OEF=23	; IDEM, EOF
207	000024	IN2E=24	; INFO, NEW BUF, CHAR2
208	000025	IN2EEF=25	; IDEM + EOF
209	000026	IB1=26	; INFO, BODY, CHAR1
210	000027	IB1EF=27	; IDEM + EOF
211	000030	IB2=30	; INFO, BODY, CHAR2
212	000031	IE2EF=31	; IDEM + EOF
213	000032	IE1=32	; INFO, END BUF, CHAR1
214	000033	IE1EF=33	; IDEM + EOF
215	000034	IE2=34	; INFO, END BUF, CHAR2
216	000035	IE2EF=35	; IDEM + EOF
217		;	
218		; COMMAND TO FIFO CONTROL	
219		;	
220	000300	BOF=300	; BEGIN OF FRAME
221	000240	EOF=240	; END OF FRAME
222	000020	FIFOTR=20	; FIFO TRACE RQ
223	000022	AMASK=22	; AND MASK
224	000021	OMASK=21	; OR MASK
225		;	
226		;	
227		;	
228		; SYSTEM PARAMETERS	
229		;	
230	000040	BLOCK=40	; BLOCK LENGTH
231	000265	BADA=265	; BAD ADDRESS
232		;	
233		;	
234		;	
235		;	
236		;	
237		; COMMAND TO PDP	
238		;	
239	000020	RDY0=20	; SET RDY0
240	000200	RDY1=200	; SET RDY1
241	000200	ITIN=200	; SET INTERRUPT IN
242	000300	ITOUT=300	; SET INTERRUPT OUT
243		;	
244		;	

```
245 ; KMC STATES
246 ;
247 ;
248 000000 IDLE =0
249 000002 WFIN =2
250 000001 WFOUT =1
251 ;
252 ;
253 ; COMMAND TYPE IN AND OUT
254 ;
255 000000 RXINFO=0
256 000002 RXLOCO=2
257 000001 RXSHCO=1
258 000003 MODSTA=3
259 000004 TXINFO=4
260 000006 TXLOCO=6
261 000005 TXSHCO=5
262 000007 COMMAN=7
263 ;
264 ;
265 ; MASKS FOR ERROR CLASSES
266 ;
267 000347 MC11=347 ; ERR 1
268 000337 MC21=337
269 000376 MC31=376
270 000075 MC41=75
271 ;
272 000020 MC12=20 ; ERR 2
273 ;
274 000377 MC13=377 ; ERR 3
275 000375 MC23=375
276 000377 MC33=377
277 000306 MC43=306
278 ;
279 ;
```



```
281 .SR7TL *KMC* INITIALISATION
282 ;
283 ;
284 ;
285 ; *****
286 ; * INITIALISATION *
287 ; *****
288 ;
289 ;
290 ;
291 ; CODE ENTERED WHEN MASTER CLEAR
292 ; IS ISSUED BY PDP11
293 ; MAR IS 0
294 ; BRG IS 0
295 ; PC IS 0
296 ; PDP IS SUPPOSED TO CLEAR CSR
297 ;
298 ;
299 ; CLEAR KMC STATUS (IDLE)
300 ;
301 004534 DEBUT: SP BR,SELB,SP7
302 ; CLEAR MAR
303 ;
304 004536 DEBUT1: MEM IMM,0,INCMAR ; CLEAR BYTE
305 004540 SP IPUS,NPR,SPO ; TEST IF MAR
306 004542 BRWRTE BR,IWCA,SPO ; OVER FLOW
307 004544 BF7 SC00
308 004546 ALWAYS DEBUT1 ; LOOP UNTIL END
309 ;
310 ;
311 ;
```

\*KMC\* MAIN LOOP

```
313 .SBTTL *KMC* MAIN LOOP
314 ;
315 ; *****
316 ; * MAIN LOOP *
317 ; *****
318 ;
319 ;
320 ; THREE TYPES OF REQUEST CAN HAPPEN
321 ; -L.U. REQUEST (IN L.U. SCAN STATUS)
322 ; -PDP REQUEST (IN CSR REGISTERS)
323 ; -OUTPUT TO PDP REQUEST (IN INTERNAL QUEUE)
324 ;
325 ; THE MAIN LOOP
326 ; SERVICES ONE L.U. (IF REQUEST)
327 ; SERVICES ONE OUTPUT TO PDP OR ONE PDP REQUEST (IF NO OUTPUT
328 ; TO DO)
329 ; THEN IT BEGINS AGAIN
330 ;
331 ;
332 ;
```

```

334 .SBTTL *KMC* SERVICING L.U. REQUESTS
335 ;
336 ;
337 ;SUCCESSFUL SCANNING
338 ; LINE NUMBER IS IN SP4
339 ;
340 ; COMPUTE BLOCK ADDRESS IN SP1
341 ; LOAD BLOCK ADDRESS IN MAR POINTER
342 ; LOAD SCAN STATUS IN SP6
343 ; AND OTHER STATUSES IN SP12 AND SP13
344 ;
345 004550 LU0: SP BR,SELB,SP6 ;SAVE SCAN REGISTER WHEN INTEREST
346 004550 ; (30 OCT 79)
347 ; FIND BLOCK ADDRESS
348 004552 SP BR,TWOA,SP1
349 004554 SP BR,TWOA,SP1
350 004556 SP BR,TWOA,SP1
351 004560 SP BR,TWOA,SP1,LDMAR
352 ;
353 ;
354 ; CLEAR DATA COUNT
355 ; TEST IF FIFO INPUT
356 ;
357 004562 BRWRTI IMM,0 ; CLEAR COUNT
358 004564 SP BR,SELB,SP14
359 004566 SP BR,SELB,SP15
360 004570 BRWRTI IMM,MFIIN
361 004572 NODST BR,AORB,SP6
362 004574 Z LUFIO
363 ;
364 ;
365 ; TEST IF FIFO OUTPUT
366 ;
367 004576 LU10: BRWRTI IMM,MFI0U
368 004600 NODST BR,AORB,SP6
369 004602 Z LUFOO
370 ;
371 ;
372 ; TEST IF STATUS INPUT
373 ;
374 004604 LU20: BRWRTI IMM,MSTIN
375 004606 NODST BR,AORB,SP6
376 004610 Z LU30
377 004612 ALWAYS SC30 ; EXIT
378 ;
379 ;
380 ; STATUS INPUT
381 ; BRANCH DEPENDING UPON CONTENTS OF
382 ; STATUS HIGH BITS 0-2
383 ;
384 004614 LU30: SP BR,AANDB,SP6
385 004616 NODST SELA,SP1,LDMAR
386 ;
387 004620 SP IBUS,RESTLO,SP12 ;GET STATUS LOW
388 004622 SP IRUS,RESTHI,SP13 ;GET STATUS HIGH
389 ;
390 004624 BRWRTI IMM,7

```

391 004626	RPWPT	ER,AANDB,SP13		
392 004630	SP	BR,SELB,SPB		
393 004632	BRADDR	LU40		
394 004634	.ALWAY	BP,ADD,SPR,PO		
395 004636	LU40:	ALWAYS LUACO	; ACK	000
396 004640		ALWAYS FS00	; EOF RX	001
397 004642		ALWAYS SC30	;	010
398 004644		ALWAYS LUFTU	; FIF TR	011
399 004646		ALWAYS LUMSO	; MOD ST	100
400 004650		ALWAYS SC30	;	101
401 004652		ALWAYS SC30	;	110
402 004654		ALWAYS LUERO	; ERROR	111
403		;		
404		;		
405		; SAVE STATUS USUALLY BEFORE EXITING		
406		; FROM A L.U. OR PDP REQUEST		
407		; ON TRANSMISSION		
408		;		
409		;		
410 004656	SC28:	NODST MEMX,SELA,SP1,LDMAR		
411 004660		MEM MEMX,SELA,SP10		
412 004662		ALWAYS LU20		

```
414 .SBTTL *KMC* SCANNING ROUTINE
415 ;
416 ;
417 ;
418 ; *****
419 ; * SCANNING ROUTINE *
420 ; *****
421 ;
422 ;
423 ; SP4 CONTAINS THE LAST PROCESSED OR SCANNED L. U.
424 ; INCREMENT SP4
425 ; LOAD LINE UNIT ADDRESS
426 ; TEST EVENT BIT (SCAN STOP)
427 ; JUMP TO SUCCESSFUL SCANNING ROUTINE IF EVENT BIT SET
428 ; ELSE SERVE PDP CSR REGISTERS
429 ;
430 ;
431 ;
432 004664 SC00: SP BR,INCA,SP4
433 004666 BRWRT IMM,7
434 004670 SPBR BR,AANDB,SP4 ; NEW LINE NUMBER
435 ;
436 004672 OUT BR,SELB,LOLUD ; LOAD LINE UNIT ADDRESS
437 004674 SP BR,SELB,SP1 ; LOAD IN SP1 AND 1 CLOCK TICK
438 004676 BRWRT IBUS,RESCAN ; READ EVENT BITS AND SAVE THEM
439 ; (30 OCT 79)
440 004700 BR7 LU0 ; JUMP IF EVENT
441 ;
442 ;
443 ;
```

```
445 .SBTTL *KMC* SERVICING OF CSR REGISTERS
446 ;
447 ;
448 ; *****
449 ; * SERVICING OF CSR REGISTERS *
450 ; * OUTPUT TO PDP OR *
451 ; * INPUT FROM PDP *
452 ; *****
453 ;
454 ;
455 ; IT IS STATE DRIVEN ROUTINE
456 ; STATE IS IN SP7
457 ; THREE POSSIBLE STATES
458 ; IDLE: THEN OUTPUT IS ALLOWED
459 ; OR IF THE QUEUE IS EMPTY
460 ; PDP ROI IS TESTED AND
461 ; ACKNOWLEDGED
462 ; WT OUTPUT: AN OUTPUT TO PDP IS
463 ; GOING ON. WHEN FINISHED,
464 ; DO AS IF IDLE STATE
465 ; WT INPUT: AN INPUT FROM PDP IS
466 ; GOING ON. WHEN FINISHED,
467 ; PDP REQUEST IS SERVICED
468 ;
469 ; TEST KMC STATE
470 ;
471 004702 SC30: BRADDR <SC31-<IDLE*2>>
472 004704 .ALWAY BR,ADD,SP7,P0
473 004706 SC31: ALWAYS SC40 ; IDLE
474 004710 ALWAYS SC50 ; WT OUTPUT
; FOR WAITING INPUT,
475
476 ;TURN PAGE
477 ;
478 ;
479 ;
```

\*KMC\* SERVICING OF PDP REQUESTS

```

481          .SBTTL *KMC*   SERVICING OF PDP REQUESTS
482          ;
483          ; *****
484          ; * SERVICING OF PDP REQUESTS *
485          ; *****
486          ;
487          ; KMC STATE IS WAITING INPUT
488          ; IS THE INPUT DONE ?
489          ; IF NO, EXIT
490          ; IF YES, SERVICIES PDP REQUEST
491          ;
492          ;
493          ;
494          ;
495 004712    BRWRT  IBUS, OCON          ; GET BSEL2
496 004714    BR7    SC00              ; INPUT NOT YET
497          ;                          ; FINISHED,
498          ;                          ; EXIT
499          ;
500          ;
501          ; INPUT FINISHED
502          ; SERVICES PDP REQUEST
503          ;
504          ; BLOCK ADDRESS IS LOADED IN MAR
505          ; DEPENDING UPON LINE NUMBER
506          ; LINE NUMBER IS IN BSEL3 BITS 0-2
507          ; SAVE LINE NUMBER IN SP4
508          ;
509 004716    P00:  SP    IBUS, LINENM, SP1
510 004720    SP    BR, T*OA, SP1
511 004722    SP    BR, T*OA, SP1
512 004724    SP    BR, T*OA, SP1
513 004726    SP    BR, T*OA, SP1, LDMAR
514 004730    SP    IBUS, LINENM, SP4
515          ;
516          ; LOAD LINE NUMBER
517 004732    OUT    IBUS, LINENM, LOLUAD
518          ;
519          ; NEW STATE = IDLE
520 004734    BRWRT  IMM, IDLE
521 004736    SP    BR, SELB, SP7
522          ;
523          ;
524          ; GET TYPE IN BSEL0 BITS 0-2
525          ; SAVE TYPE IN + PS + FIFOE IN SP2
526          ; BRANCH TO SPECIALIZED ROUTINE
527          ; DEPENDING UPON TYPE IN
528          ; IF COMMAND THEN BRANCH TO
529          ; SPECIALIZED ROUTINE DEPENDING
530          ; UPON COMMAND TYPE (IN BSEL6)
531          ;
532          ;
533          ;
534 004740    SPBF   IBUS, INCUN, SP2    ; READ BSEL0
535 004742    SP    BR, SELB, SP0
536 004744    BRWRT  IMM, 7
537 004746    SP    RR, AANDB, SP0
    
```

\*KMC\* SERVICING OF PDP REQUESTS

538 004750	BRADDR P10	
539 004752	.ALWAY BR,ADD,SP0,P0	
540 004754	P10: ALWAYS PJ00	; TABLE AD FOR RX
541 004756	ALWAYS P20	; INVALID COMMAND
542 004760	ALWAYS P20	
543 004762	ALWAYS PK00	; RQ FOR MODEM STATUS
544 004764	ALWAYS PH00	; TX OF INFO FRAME
545 004766	ALWAYS PD00	; TX OF SHORT CONTROL FRAME
546 004770	ALWAYS PC00	; TX OF LONG CONTROL FRAME
547	;	
548 004772	SP 1BUS,PORT3,SP3	; COMMAND
549 004774	BRWPTE IMM,16	
550 004776	COMP BR,SP3	
551 005000	Z PF00	; ABORT TRANSMISSION (16)
552 005002	C PG00	; ABORT RECEPTION (17)
553		; AND DUMMY COMMAND (20)
554 005004	BRWPTE IMM,1	
555 005006	COMP BR,SP3	
556 005010	C PH00	; MISCELLANEOUS COMMANDS(2 TO 15)
557 005012	Z PE00	; LOAD MASK (1)
558 005014	ALWAYS PA00	; BASE ADDRESS (0)



```
560 .SBTTL *KMC* WT OUTPUT
561 ;
562 ;
563 ; *****
564 ; * WAITING OUTPUT *
565 ; *****
566 ;
567 ;
568 ; KMC STATE IS WT OUTPUT
569 ; IF OUTPUT FINISHED, BRANCH TO
570 ; SERVICING OF OUTPUT TO PDP
571 ; IF NOT, EXIT
572 ;
573 ;
574 ;
575 005016 SC50: BRWRIE IRUS,OCON ; GET BSEL2
576 005020 BR4 SC00 ; NOT FINISHED
577 ; FINISHED
578 ;
```

\*KMC\* SERVICING OF OUTPUT TO PDP

```

580 .SBTTL *KMC* SERVICING OF OUTPUT TO PDP
581 ;
582 ;
583 ; *****
584 ; * SERVICING OF OUTPUT TO PDP *
585 ; *****
586 ;
587 ; KMC STATE IS IDLE
588 ; TEST IF OUTPUT TO BE DONE
589 ; IF YES, OUTPUT MESSAGE AND UPDATE
590 ; QUEUE
591 ; NEW KMC STATE IS WT OUTPUT
592 ; IF NO, TEST IF REQUEST FROM PDP
593 ; IF YES, ACKNOWLEDGE REQUEST
594 ; NEW KMC STATE IS WT INPUT
595 ; IF NO, STAY IDLE
596 ;
597 ;
598 ;
599 ;
600 ;1-TEST OUTPUT TO PDP
601 ;
602 005022 SC40: LDMAP IMM,2 ; POINTER ON LAST
603 005024 LDMA IMM,2776 ; MESSAGE
604 005026 BPWRITE MEMX,SELB,INCMAR
605 005030 SP MEMX,SELB,SP1 ; POINTER ON FIRST
606 ; MESSAGE
607 005032 COMP BR,SP1 ; TEST IF NO MESS
608 005034 Z SC42 ; EXIT
609 ;
610 005036 BRWRITE IMM,10 ; NOW POINTS ON
611 005040 MEM BR,ADD,SP1 ; MESSAGE TO
612 ; SEND TO PDP
613 ;
614 ;2 - NPR OUTPUT
615 005042 SC410: LDMAP IMM,2 ; FIND BAD
616 005044 LDMA IMM,BADA
617 ;
618 005046 OUT MEMX,SELB,ORA1,INCMAR
619 005050 OUT MEMX,SELB,ORA2,INCMAR
620 005052 SPBR MEMX,SELB,SP0
621 005054 BRO SC409 ; IF NO BAD, NPR IS
622 ; NOT PERFORMED
623 005056 ALWAYS SC415
624 005060 SC409: OUT BR,DECA,OBR
625 ;
626 005062 NODST BR,SELA,SP1,LDMAR ; POINT ON
627 ; MESSAGE
628 005064 SP IMM,3,SP3,LDMAPG ; INIT LOOP
629 005066 SC411: OUT MEMX,SELB,OUTDA1,INCMAR
630 005070 OUT MEMX,SELB,OUTDA2,INCMAR
631 005072 BPWRITE IMM,21
632 005074 OUT BR,SELB,ONPR ; RO DMA
633 005076 SC412: BRWRITE IBUS,NPR
634 005100 BRO SC412
635 005102 SP BR,DECA,SP3
636 005104 Z SC415 ; END, EXIT

```

```

637 005106          CALLSR  SP5,RXDMA1,SCREL          ; INC ADDRESS
638
639
640 005114          SC415: BRWRTE IMM,2,LDMAPG          ; POINT ON
641 005116          NODST  BR,INCA,SP1,LDMAR          ; MESSAGE
642
643 005120          OUT    MFMX,SELB,OPORT2,INCMAR
644 005122          OUT    MFMX,SELB,OPORT1,INCMAR
645 005124          OUT    MFMX,SELB,OLINEN,INCMAR
646 005126          OUT    MFMX,SELB,OOCON
647
648
649 005130          ;
650 005132          BRWRTE IBUS,INCON          ; SEE IF INT SET
651
652 005134          SC416: BRWRTE IMM,WTOUT          ; NEW KMC
653 005136          SC417: SP    BR,SELB,SP7          ; STATE
654
655 005140          ALWAYS  SC00          ; EXIT
656
657 005142          SC418: BRWRTE IMM,ITOUT          ; DEAL WITH INT
658 005144          OUT    BR,SELB,OBRR
659 005146          ALWAYS  SC416
660
661
662
663
664
665
666
667
668 005150          SC42:  SPBR  IBUS,INCON,SPO          ; GET BSEL0
669 005152          BR7    SC43
670
671 005154          BRWRTE IMM,IDLE          ; NO ROI
672
673 005156          ALWAYS  SC417          ; STATE=IDLE
674
675 005160          SC43:  BRWRTE IMM,0          ; REQUEST
676 005162          OUT    BR,SELB,OLINEN          ; CLEAR BSEL3 AS
677 005164          BRWRTE IMM,RDYI          ; REQUESTED BY PD
678 005166          OUT    BR,SELB,OOCON          ; SET RDYI
679 005170          BRWRTE BR,TWOA,SPO          ; SEE IF INT SET
680 005172          BR7    SC44
681
682
683 005174          SC430: BRWRTE IMM,W11N          ; STATE=WAIT IN
684 005176          ALWAYS  SC417          ; EXIT
685
686 005200          SC44:  BRWRTE IMM,ITIN          ; IT GENERATED
687 005202          OUT    BR,SELB,OBRR
688 005204          ALWAYS  SC430

```



\*KMC\* FIFO INPUT

```

751 005264          ALWAYS  F10200          ; WT CHARACTER ADDRESS
752 005266          ALWAYS  F10300          ; IDEM + EOF
753 005270          ALWAYS  F10400          ; LONG FRAME, DATA 1
754 005272          ALWAYS  F10500          ; IDEM + EOF (ERROR)
755 005274          ALWAYS  F10600          ; LONG FRAME, DATA 2
756 005276          ALWAYS  F10500          ; IDEM + EOF (ERROR)
757 005300          ALWAYS  F10800          ; LONG FRAME, DATA 3
758 005302          ALWAYS  F10900          ; IDEM +EOF
759 005304          ALWAYS  F11000          ; ABORT RX
760 005306          ALWAYS  F11100          ; IDEM + EOF
761 005310          ALWAYS  F11000          ; ABORT REQUESTED BY PDP
762 005312          ALWAYS  F11300          ; IDEM +EOF
763 005314          ALWAYS  F11200          ; INFO FRAME, NO BUFFER
764 005316          ALWAYS  F11200          ; IDEM + EOF
765 005320          ALWAYS  F11400          ; NEW BUF, CHAR 1, EV
766 005322          ALWAYS  F11500          ; IDEM + EOF
767 005324          ALWAYS  F11600          ; NEW BUF, CHAR 1, ODD
768 005326          ALWAYS  F11700          ; IDEM + EOF
769 005330          ALWAYS  F12210          ; NEW BUF, CHAR 2,EVEN
770 005332          ALWAYS  F11900          ; IDEM + EOF
771 005334          ALWAYS  F12000          ; CHAR 1
772 005336          ALWAYS  F12100          ; IDEM + EOF
773 005340          ALWAYS  F12200          ; CHAR 2
774 005342          ALWAYS  F12300          ; IDEM + EOF
775                ;
776                ;
777 005344          LUF150: BR0      LUF152          ; ERROR
778                ;
779 005346          LUF151: SP      BR,INCA,SP10      ; END OF FRAME,
780 005350          BRWRTE IMM,34
781 005352          SP      BR,AANDB,SP11
782 005354          ALWAYS  LUF130
783                ;
784                ;
785                ; NOW, ERROR CASES
786                ;
787                ;
788 005356          LUF152: OUT     IBUS,REFIDA,OBA1    ; ERROR CASE
789                ;
790                ; SET THE FLAG : NO BUFFER AVAILABLE
791                ; WHEN - STATE = INFO FRAME, NO BUFFER
792                ; - STATE = FREE (ERROR ON FIRST CHARACTER)
793                ; - STATE = WAITING CHARACTER ADDRESS (ERROR ON 2ND CHAR)
794                ;
795 005360          BRWRTE IMM,INOB
796 005362          COMP   BR,SP10
797 005364          Z      LUF153          ; INFO FRAME, NO BUFFER
798                ;
799 005366          BRWRTE IMM,CTRLEF          ; 1ST OR 2ND CHAR
800 005370          COMP   BR,SP10
801 005372          C      F11110
802                ;
803 005374          LUF153: BRWRTE IMM,17          ; NO BUFFER AVAILABLE
804 005376          MODST  BR,ADD,SP1,LDMAR
805 005400          MEM    IMM,NOBUFF
806 005402          ALWAYS  F11120

```

```

808 ;
809 ;
810 ; END OF ROUTINES
811 ; -----
812 ;
813 ; 1 - NON INFORMATION FRAMES
814 ;
815 ; SAVE NEW STATE IN SP10 AND BRANCH TO TEST FIFO
816 ;
817 005404 LUF157: SP ER,SELB,SP10
818 005406 ALWAYS LUF162
819 ;
820 ;
821 ;
822 ;
823 ; 2 - INFORMATION FRAMES
824 ;
825 ; SAVE NEW STATE IN SP10
826 ; DECREMENT COUNT
827 ; IF COUNT = ZERO, BRANCH TO FIND NEW BUFFER
828 ; TEST FIFO : IF NOT READY, SAVE CONTEXT AND EXIT
829 ; ELSE, GO TO DISPATCHING ROUTINE
830 ;
831 ;
832 ;
833 ;
834 005410 LUF160: SP BR,SELB,SP10 ; LOAD NEW STATE
835 005412 LUF161: SP BR,DECA,SP14 ; DECREMENT COUNT
836 005414 Z LUF168
837 ;
838 005416 LUF163:
839 ;;SP BR,DECA,SP9 ; DEC BLOCK COUNT
840 ;;Z LUF164 ; JUMP IF END
841 ;
842 005416 LUF162: SP IBUS,RESCAN,SP8 ; READ FIFO SCAN
843 005420 BRWRTE BR,TWOA,SP8
844 005422 BR7 LUF120 ; BRANCH FIFO READY
845 ;
846 005424 LUF164:
847 005424 NODST BR,SELA,SP1,LDMAR ; NOT READY, SAVE
848 005426 MEM MEMX,SELA,SP10 ; CONTEXT
849 005430 BRWRTE IMM,11
850 005432 NODST BR,ADD,SP1,LDMAR
851 005434 MEM IBUS,INDAT2,INCMAR
852 005436 MEM MEMX,SELA,SP14,INCMAR
853 005440 MEM MEMX,SELA,SP15,INCMAR
854 005442 MEM IBUS,IOPA1,INCMAR
855 005444 MEM IBUS,IOPA2,INCMAR
856 005446 SP IBUS,UBBR,SPO ; SAVE HIGH ADDRESS
857 005450 MEM IMM,14
858 005452 MEM MEMX,AANDB,SPO,INCMAR
859 ;
860 005454 LUF165: BRWRTE IBUS,NPR ; WAIT DMA IF ANY RUNNIN
861 005456 BR0 LUF165
862 005460 BRWRTE IBUS,UBBR
863 005462 BR0 FI1540 ; NON EX MEM
864 005464 ALWAYS LU10 ; EXIT
    
```

865		;	
866		; DECREMENT COUNT	
867		;	
868 005466	LUF168: SP	BR,DECA,SP15	; COUNT HIGH
869 005470	Z	LUF169	
870 005472	ALWAYS	LUF163	
871		;	
872 005474	LUF169: BRWPE	1MM,1B2	; LAST CHAR
873 005476	COMP	BR,SP10	; OF THE BUFFER
874 005500	Z	F12400	; DMA TO PERFORM
875 005502	ALWAYS	F12410	; NO DMA

```
877 .SBTTL *KMC* ASK FOR STATUS
878 ;
879 ; *****
880 ; * ASK FOR MODEM STATUS *
881 ; *****
882 ;
883 ; TEST IF UNIT ON LINE
884 ; SEND TYPE OF STATUS IN LOW
885 ; SEND REQUEST IN HIGH
886 ; EXIT
887 ;
888 ;
889 ;
890 005504 PK00: BRWRTI IRUS,RELUAD ; CHECK IF ON LINE
891 005506 BR4 PK01 ; ON LINE
892 005510 ALWAYS TXIN11 ; OFF LINE , HORROR
893 005512 PK01: BRWRTI IMM,4 ; STATUS DEMAND
894 005514 OUT BR,SELB,LOSTLO
895 005516 BRWRTI IMM,24
896 005520 OUT BR,SELB,LOSTHI
897 005522 ALWAYS SC30 ; EXIT
```



```
899 .SETTL *KMC* TABLE ADDRESS FOR RECEPTION
900 ;
901 ; *****
902 ; * TABLE ADDRESS FOR RECEPTION *
903 ; *****
904 ;
905 ; PAGE 1
906 ; LOAD RX STATUS IN SP10
907 ; TABLE ADDRESS IS IN BSEL4,5,7
908 ; CHECKS VALIDITY OF REQUEST
909 ; SAVE TABLE ADDRESS IN MAR
910 ; SET TABLE ADDRESS FLAG IN MAR+3
911 ;
912 ; EXIT
913 ;
914 005524 PJ00: BRWRTI IMM,1
915 005526 SP BR,SELB,SPO,LDMAPG
916 ;
917 005530 SP MEMX,SELB,SP10 ; RX STATUS
918 005532 BRWRTI IMM,1NOBLF
919 005534 COMP BR,SP10
920 005536 C TXINI2 ; BUSY, EXIT
921 ;
922 005540 CALLSB SP5,BADIN ; READ BAD
923 005546 CALLSB SP5,H1PAAD
924 005554 MEM MEMX,INCA,SP8
925 005556 ALWAYS SC30
926 ;
```

```
928  
929 .SBTTL *KMC* INVALID COMMAND  
930 ;  
931 ; *****  
932 ; * INVALID COMMAND *  
933 ; *****  
934 ;  
935 ; QUEUE AN ERROR MESSAGE  
936 ;  
937 005560 P20: BRWTE IMM,INVCOM  
938 005562 ALWAYS ERKMC  
939 ;
```

```

941 .SBTIL *KMC* TX OF INFO FRAME
942 ;
943 ; *****
944 ; * TRANSMISSION OF INFORMATION FRAME *
945 ; *****
946 ;
947 ; 1 - INITIALISATION
948 ; -----
949 ;
950 ; CALL ROUTINE THAT DOES
951 ; -LOAD PAGE 0
952 ; -CHECKS VALIDITY OF REQUEST
953 ; CALL ROUTINE THAT READS BAD AND LOAD
954 ; -TWO FIRST CHARACTERS IN MAR
955 ; -TWO LAST CHARACTERS IN NPR
956 ; CALL ROUTINE THAT HANDLE HIGH ADDRESS
957 ; AND LOADS IT IN MAR + 3
958 ;
959 ; INIT OVERALL COUNT TO ZERO (MAR + 4 AND 5)
960 ; NEW STATE IS BEGIN OF FRAME
961 ;
962 ; LOAD BEGIN OF FRAME IN FIFO CONTROL
963 ; LOAD TYPE IN MAR + 6
964 ; LOAD FIFOE IN MAR + 7, WHEN REQUESTED
965 ; STORE TWO FIRST CHARACTERS IN MAR + 8 AND 9
966 ;
967 ; EXIT TO DISPATCHING POINT
968 ;
969 ;
970 005564 PB00: CALLSB SP5,TXINI
971 005572 CALLSB SP5,BADIN
972 005600 CALLSB SP5,HIPAAD
973 005606 MEM MEMX,INCA,SP8
974 ;
975 005610 BRWRTE IMM,BEGFRA ; NEW STATE
976 ;
977 005612 PB0010: SP BR,SELB,SP10,INCMAR
978 005614 MEM IMM,0,INCMAR ; OVERALL
979 005616 MEM IMM,0,INCMAR ; COUNT TO 0
980 ;
981 005620 BRWRTE IMM,BOF ; BEGIN OF FRAME IN FIFO CTRL
982 005622 OUT BR,SELB,LOFICO
983 005624 MEM IMM,57
984 005626 MEM MEMX,AANDB,SP2,INCMAR ; SAVE TYPE IN + FIFOE+PS, IN MAR
985 ;
986 005630 BRWRTE IMM,40 ; GET FIFOE ONLY
987 005632 BRWRTE BR,AANDB,SP2
988 005634 BROTAT ; PREPARE FOR LAST FIFO CONTROL
989 005636 MEM BR,SELB,INCMAR ; LOAD IN MAR
990 ;
991 005640 MEM IBUS,PORT1,INCMAR ; FIRST CHAR
992 005642 MEM IBUS,PORT2 ; 2ND CHAR
993 ;
994 005644 ALWAYS TX00
995 ;
    
```

```
997 ;  
998 ;  
999 ; *****  
1000 ; * TRANSMISSION OF INFORMATION FRAME *  
1001 ; *****  
1002 ; 2 - SEND ADDRESS CHARACTERS  
1003 ; -----  
1004 ;  
1005 ; THIS ROUTINE IS ENTERED WHEN TX STATUS IS BEGIN OF FRAME  
1006 ;  
1007 ; SEND 10 FIFO ADDRESS CHAR (MAR + 9)  
1008 ; LOAD IN NPR3, CONTROL CHAR  
1009 ; LOAD BODY OF FRAME IN FIFO CTRL  
1010 ; INIT BLOCK COUNTER IN SP9  
1011 ;  
1012 ;  
1013 005646 PBA00: BRWRTI IMM,10 ; MAR ON DATA  
1014 005650 NODST BR,ADD,SP1,LDMAR  
1015 ;  
1016 005652 OUT MEMX,SELB,LOFIDA,INCMAR  
1017 005654 OUT MEMX,SELB,OUTDA2 ; FIFO DATA  
1018 ;  
1019 005656 BRWRTI IMM,0  
1020 005660 OUT BR,SELB,LOFICO ; BODY OF FRAME  
1021 005662 BRWRTI IMM,BLOCK  
1022 005664 SP BR,SELB,SP9 ; BLOCK COUNT
```

```

1024 ;
1025 ; *****
1026 ; * TRANSMISSION OF INFORMATION FRAME *
1027 ; *****
1028 ;
1029 ;
1030 ; 3 - FIRST DMA TO GET CHARACTER ADDRESS
1031 ; -----
1032 ;
1033 ; CALL ROUTINE TO EXECUTE DMA
1034 ; TEMPORARILY SAVE CHARACTER ADDRESS
1035 ; LOW AND MEDIUM IN SPO AND NPR7
1036 ; NEW STATE IS COMPUTED DEPENDING UPON
1037 ; PARITY OF ADDRESS LOW
1038 ;
1039 ;
1040 005666 PBA10: BRADDR FBD98 ; ERROR AD
1041 005670 SP BR,SELB,SP11
1042 005672 CALLSR SP8,TABAD,TABAR1 ;DMA
1043 005700 PBA15: OUT IBUS,INDAT2,OBA2 ; SAVE DATA
1044 005702 SPBR IBUS,INDAT1,SPO
1045 005704 BR0 PBA20
1046 005706 BRWRTE IMM,FICHR ; NEW STATE
1047 005710 ALWAYS PBA30 ; IF EVEN ADDR
1048 005712 PBA20: SP BR,DECA,SPO ; MAKE ADDRESS EVEN
1049 005714 BRWRTE IMM,BEBUOD ; IF ODD ADDR
1050 005716 PBA30: SP BR,SELB,SP10
    
```

```
1052 ;
1053 ; *****
1054 ; * TRANSMISSION OF INFORMATION FRAME *
1055 ; *****
1056 ;
1057 ;
1058 ;
1059 ; 4 - SECOND DMA TO GET CHARACTER COUNT
1060 ; -----
1061 ;
1062 ; CALL ROUTINE THAT DOES :
1063 ; CALL ROUTINE TO PERFORM DMA
1064 ; THIS ALLOWS TO GET COUNT LOW
1065 ;                            COUNT HIGH
1066 ;                            LAST BUFFER FLAG
1067 ;                            CHAR ADDRESS HIGH
1068 ; COUNT LOW AND HIGH IN SP14 AND 15
1069 ; IF LAST BUFFER, BIT COUNT IN MAR + 7
1070 ; CHAR ADDRESS HIGH IN BRG
1071 ;
1072 ; LOAD ADDRESS HIGH IN NPR2
1073 ; LOAD SAVED ADDRESS LOW AND MEDIUM IN
1074 ;    NPR4 AND NPR5
1075 ;
1076 ; NOW, THE NORMAL TRANSFER OF DATA FROM
1077 ; PDP MEMORY TO LINE UNIT CAN START
1078 ;
1079 005720 CALLSB SP5,DMAC
1080 005726 MEM    MEMX,AORB,SP3
1081 005730 BRWRTE BR,INCA,SP8
1082 005732 OUT    BR,SELB,OUTDA1            ; ADDR HIGH
1083 005734 OUT    IBUS,IOBA2,IBA2           ; ADDR MEDIUM
1084 005736 OUT    SELA,IBA1                 ; ADDR LOW
```

```

1086 ;
1087 ; *****
1088 ; * TRANSMISSION OF INFORMATION FRAME *
1089 ; *****
1090 ;
1091 ;
1092 ;
1093 ; 5 - DATA TRANSFER TO L.U.
1094 ; -----
1095 ;
1096 ; REQUEST 2 DATA THRU DMA
1097 ; EXIT, IF FIFO NOT FREE
1098 ; EXIT, IF END OF BLOCK
1099 ; REQUEST NEW BUFFER IF END OF BUFFER
1100 ; OR SEND END OF FRAME TO FIFO CONTROL
1101 ; IF LAST BUFFER. IN THIS CASE, ALSO
1102 ; DEALS WITH FIFOE
1103 ; SEND PREVIOUSLY SAVED DATA TO FIFO
1104 ;
1105 ; THEN IT IS STATE DRIVEN
1106 ; STATE: BEGIN OF BUFFER ODD
1107 ; NEW STATE = 1ST CHARACTER
1108 ; WAIT FOR END OF DMA
1109 ; TEST MEMORY INEXISTENT
1110 ; SAVE DATA HIGH
1111 ; INCREMENT CHARACTER ADDRESS
1112 ; GO TO TRANSFER DMA
1113 ;
1114 ;
1115 ; STATE: 1ST CHARACTER
1116 ; NEW STATE = 2ND CHARACTER
1117 ; WAIT FOR END OF DMA
1118 ; TEST MEMORY INEXISTENT
1119 ; SAVE DATA LOW
1120 ; GO TO TEST FIFO FREE
1121 ; STATE: 2ND CHARACTER
1122 ; NEW STATE = 1ST CHARACTER
1123 ; SAVE DATA HIGH
1124 ; INCREMENT CHARACTER ADD RESS
1125 ; GO TO TRANSFER DMA
1126 ;
1127 ;
1128 005740 PBC00: OUT IBUS,IODAT1,ONPR ; START DMA
1129 005742 PBC10: BRWRT 1BUS,RESCAN ; FIFO FREE?
1130 005744 BRPTAT
1131 005746 BR4 PBD00 ;
1132 005750 PBC20: INCA ; FIFO NOT FREE
1133 ; SAVE CONTEXT
1134 005752 MEM IBUS,IODAT2,INCMAR ; DATA SAVED
1135 005754 MEM IBUS,INDAT2,INCMAR ; DATA HIGH
1136 005756 MEM MEMX,SELA,SP14,INCMAR ; COUNT
1137 005760 MEM MEMX,SELA,SP15,INCMAR
1138 005762 MEM IBUS,IIRA1,INCMAR ; CHAR ADDR
1139 005764 MEM IBUS,IIBA2,INCMAR
1140 005766 MEM IBUS,IODAT1
1141 005770 ALWAYS SC28 ; TX STATUS
1142 ; AND EXIT
    
```

```

1143
1144 005772 PBD00: SP BR,DECA,SP9 ; DEC BLOCK
1145 005774 Z PFC20 ; COUNT1, TEST
1146 005776 SP BR,DECA,SP14 ; DEC LOW COUNT
1147 006000 Z PBD00
1148
1149
1150 006002 PBD10: OUT IBUS,IODAT2,LOFIDA ; OUTPUT DATA
1151 006004 BRADDR <PBD20-<REBUOD*2>> ; TEST STATUS
1152 006006 .ALWAY BR,ADD,SP10,P1
1153 006010 PBD20: ALWAYS PBD40 ; ODD BUFFER
1154 006012 ALWAYS PBD50 ; 2ND CHAR
1155 ; 1ST CHAR
1156
1157 006014 SP BR,DECA,SP10 ; NEW STATE
1158 006016 PBD30: BRWRTE IBUS,NPR ; WAIT DMA
1159 006020 BRO PBD30
1160 006022 BRWRTE IBUS,UBBR ; MEM INEX?
1161 006024 BRO PBD98
1162 006026 OUT IBUS,INDAT1,OUTDA2 ; SAVE LOW DATA
1163 006030 ALWAYS PBC10
1164
1165
1166 006032 PBD40: BRWRTE IBUS,NPR ; ODD BUFFER
1167 006034 BRO PBD40 ; WAIT DMA
1168 006036 BRWRTE IBUS,UBBR ; MEM INEX?
1169 006040 BRO PBD98
1170
1171 006042 PBD50: BRWRTE IMM,FICHAR ; 2ND CHAR
1172 006044 SP BR,SELB,SP10 ; NEW STATE
1173 006046 OUT IBUS,INDAT2,OUTDA2 ; SAVE HIGH DATA
1174 006050 SP IBUS,IIBA1,SPO ; INC ADDRESS
1175 006052 BR,INCA,SPO ; LOW
1176 006054 OUT BR,INCA,IBA1
1177 006056 C PBD60 ; INC ADD HIGH
1178 006060 ALWAYS PBC00
1179
1180 006062 PBD60: SP IBUS,IIBA2,SPO ; INC AD MEDIA
1181 006064 OUT BR,APLUSC,IBA2
1182 006066 C PBD70
1183 006070 ALWAYS PBC00
1184
1185 006072 PBD70: BRWRTE IMM,4 ; INC AD HIGH
1186 006074 SP IBUS,IODAT1,SPO
1187 006076 OUT BR,ADD,OUTDA1
1188 006100 ALWAYS PRC00
1189
1190
1191
1192 006102 PBD80: SP BR,DECA,SP15 ; DEC HIGH COU
1193 006104 Z PBD90
1194 006106 ALWAYS PBD10
1195
1196 006110 PBD90: BRWRTE MEMX,SELB ; END OF BUFFER
1197 006112 BR7 PBD95 ; LAST BUFFER?
1198 006114 ALWAYS PBA10 ; GET NEW BUFFER
1199

```



1200  
1201  
1202  
1203 006116  
1204 006120  
1205 006122  
1206 006124  
1207  
1208 006126  
1209 006130  
1210

```
      ;  
      ;  
      ; LAST BUFFER  
PBD95: OUT BR,SELB,LOFICO ; LOAD FIFO CO  
      OUT IEUS,IODAT2,LOFIDA ; OUTPUT DATA  
      BR4 SC2R ; FIFOE, EXIT  
      ALWAYS OKMES ; MESSAGE OK  
      ;  
PBD98: BRWTF IMM,TXNOEM ; ERROR  
      ALWAYS TX000  
      ;
```

\*KMC\* TX DISPATCHING ROUTINE

```

1212          .SBTTL *KMC* TX DISPATCHING ROUTINE
1213          ;
1214          ; *****
1215          ; * TX DISPATCHING ROUTINE *
1216          ; *****
1217          ;
1218          ; THIS ROUTINE IS ENTERED WHENEVER
1219          ; FIFO FREE HAS TO BE TESTED
1220          ; IF FIFO IS NOT FREE,
1221          ; THEN EXIT
1222          ; ELSE IT IS A JUMP TO SPECIALIZED
1223          ; ROUTINE, DEPENDING UPON TX STATE
1224          ; WITH MAR INITIALIZED TO BLOCK
1225          ; ADDRESS
1226          ;
1227          ;
1228 006132     TX000: SP      BR,SELB,SP10          ; LOAD NEW STATE
1229 006134     TX000: BRWRTE IBUS,RESCAN          ; CHECK FIFO FREE
1230 006136     BR0TAT
1231 006140     BR4      LUFO10                    ; DISPATCH
1232 006142     ALWAYS  SC28                      ; EXIT,FIFO FULL
1233          ;
1234          ;
1235          ;
1236          ;
1237 006144     DMACR: ALWAYS FI1245              ; RELAY ADDRESS IN PAGE 1
1238          ;                                  ; FOR END OF SUBROUTINE DMAC
1239          ;
1240          ;
1241          ;

```

```
1243 .SBTTL *KMC* TX OF LONG CONTROL FRAME
1244 ;
1245 ;
1246 ; *****
1247 ; * TX OF LONG CONTROL FRAME *
1248 ; *****
1249 ;
1250 ; 1 - INITIALISATION
1251 ; -----
1252 ;
1253 ; CALL ROUTINE THAT DOES
1254 ; -LOAD PAGE 0
1255 ; -CHECKS VALIDITY OF REQUEST
1256 ; CALL ROUTINE THAT DOES
1257 ; -READS BAD
1258 ; -SAVES TWO FIRST CHAR IN MAR
1259 ; SAVE LAST CHAR IN MAR + 3
1260 ;
1261 ; NEW STATE IS LONG FRAME, ADDRESS
1262 ;
1263 ; BRANCH TO END OF INIT ROUTINE
1264 ;
1265 006146 PC00: CALLSB SP5, TXIN1
1266 006154 CALLSB SP5, BADIN
1267 ;
1268 006162 MEM IBUS, INDAT1
1269 ;
1270 006164 BRWRT IMM, LFADDR
1271 ;
1272 006166 ALWAYS PB0010
1273 ;
1274 ;
```

```
1276 ;  
1277 ; *****  
1278 ; * TX OF LONG CONTROL FRAME *  
1279 ; *****  
1280 ;  
1281 ; 2 - SEND ADDRESS CHARACTER  
1282 ; -----  
1283 ;  
1284 ; THIS ROUTINE IS ENTERED WHEN STATUS IS LONG FRAME, ADDRESS  
1285 ; SEND ADDRESS CHARACTER TO FIFO DATA  
1286 ; LOAD BODY OF FRAME IN FIFO CTRL  
1287 ;  
1288 ;  
1289 006170 PCA00: BRWRT IMM,10 ; INIT MAR  
1290 006172 NOUST BR,ADD,SP1,LDMAR  
1291 ;  
1292 006174 OUT MEMX,SELB,LOFIDA ; FIFO DATA  
1293 006176 BRWRT IMM,0  
1294 006200 OUT BR,SELB,LOFICO ; FIFO CTRL  
1295 ;  
1296 006202 BRWRT IMM,LFCTRL  
1297 006204 ALWAYS TX000  
1298 ;  
1299 ;
```

```
1301 ;
1302 ;
1303 ; *****
1304 ; * TX OF LONG CONTROL FRAME *
1305 ; *****
1306 ;
1307 ; 3 - OUTPUT CONTROL CHARACTER
1308 ; -----
1309 ;
1310 ; INITIATE MAR TO BLOCK + 9
1311 ; OUTPUT DATA
1312 ; GO TO DISPATCHING ROUTINE
1313 ;
1314 006206 PC800: BPWRTI IMM,11 ; INIT MAR
1315 006210 NODST BR,ADD,SP1,LDMAR ; OUTPUT DATA
1316 006212 OUT MEMX,SELB,LOFIDA ; INIT MAR
1317 006214 NODST BR,INCA,SP1,LDMAR ; NEW STATE
1318 006216 BRWRTI IMM,LFDAT1
1319 006220 ALWAYS TX000
1320 ;
1321 ;
```

\*KMC\* TX OF LONG CONTROL FRAME

```
1323 ;
1324 ; *****
1325 ; * TX OF LONG CONTROL FRAME *
1326 ; *****
1327 ;
1328 ; 4 - OUTPUT DATA 1 OR 2
1329 ; -----
1330 ;
1331 ; INCREASE MAR
1332 ; SEND OUT DATA TO FIFO
1333 ; DECREMENT STATE
1334 ; GO TO DISPATCHING ROUTINE
1335 ;
1336 ;
1337 006222 PCC00: BRWRT BR,DECA,SP10 ; DEC TX STATUS
1338 006224 OUT MEMX,SELB,LOFIDA ; OUTPUT DATA
1339 006226 ALWAYS TX000
```

```

1341 ; *****
1342 ; * TX OF LONG CONTROL FRAME *
1343 ; *****
1344 ;
1345 ;
1346 ; 5 - SEND LAST CHARACTER
1347 ; -----
1348 ;
1349 ;
1350 ; THIS ROUTINE IS ENTERED WHEN
1351 ; TX STATUS IS LONG FRAME, DATA 3
1352 ;
1353 ; LOAD LAST FIFO CONTROL CHARACTER
1354 ; OUTPUT DATA 3
1355 ; DEPENDING UPON FIFOE, EXIT STRAIGHT
1356 ; OR SEND A MESSAGE TO PDP
1357 ;
1358 006230 PCD00: BRWRT  IMM,3 ; MAR ON DATA 3
1359 006232 MODST  BR,ADD,SP1,LDMAR ; SAVE DATA 3
1360 006234 OUT    MEMX,SELB,OUTDA2,INCMAR
1361 006236 BRWRT  IMM,EOF,INCMAR
1362 006240 SP     BR,SELB,SPO,INCMAR
1363 006242 INCMR
1364 006244 BRWRT  MEMX,AORB,SPO ; POINT ON FIFOE
1365 006246 ALWAYS PED95 ; END OF FRAME
1366 ;
1367 ;
    
```

```
1369 .SBTIL *KMC* 1X OF SHORT CONTROL FRAME
1370 ;
1371 ; *****
1372 ; * TX OF SHORT CONTROL FRAME *
1373 ; *****
1374 ;
1375 ;
1376 ; 1 - INITIALISATION
1377 ; -----
1378 ;
1379 ; CALL ROUTINE THAT DOES
1380 ; -LOADS PAGE 0
1381 ; -CHECKS VALIDITY OF REQUEST
1382 ; POINTS PAR ON MAR + 3
1383 ; NEW STATE IS SHORT FRAME, ADDR
1384 ;
1385 ; BRANCH TO END OF INIT ROUTINE
1386 ;
1387 ;
1388 006250 PD00: CALLSB SPS,TXINI
1389 ;
1390 006256 BPWRITE IMM,3
1391 006260 NOUST EP,ADD,SP1,LDMAR
1392 ;
1393 006262 BRWRITE IMM,SFADDR ; NEW STATE
1394 ;
1395 006264 ALWAYS FB0010
1396 ;
1397 ;
```



1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415 006266  
1416 006270  
1417  
1418 006272  
1419  
1420 006274  
1421 006276  
1422  
1423

```
;  
; *****  
; * TX OF SHORT CONTROL FRAME *  
; *****  
;  
; 2 - SEND ADDRESS CHARACTER  
; -----  
; SEND OUT ADDRESS CHARACTER  
; NEW STATE = SHORT FRAME, CTRL  
;  
;  
;  
PDA00: BRWRTF IMM,10  
; NODST BR,ADD,SP1,LDMAR  
; OUT MEMX,SELB,LOFIDA ; FIFO DATA  
;  
; BRWRTF IMM,SFCTRL ; NEW STATE  
; ALWAYS TX000 ; EXIT  
;  
;
```

```

1425 ;
1426 ;
1427 ; *****
1428 ; * TX OF SHORT CONTROL FRAME *
1429 ; *****
1430 ;
1431 ; 3 - SEND LAST CHARACTER
1432 ; -----
1433 ;
1434 ; THIS ROUTINE IS ENTERED WHEN
1435 ; TX STATUS IS SHORT FRAME, CTRL
1436 ;
1437 ; POINT MAR ON BLOCK + 7
1438 ; LOAD FIFO CTRL
1439 ; OUTPUT DATA
1440 ; DEPENDING UPON FIFOE, EXIT STRAIGHT
1441 ; OR SEND A MESSAGE TO PDP
1442 ;
1443 006300 PDB00: BRWTE IMM,7 ; COMPUTE MAR
1444 006302 NODST BE,ADD,SP1,LDMAR
1445 006304 BRWTE IMM,EOF ; GET FIFOE
1446 006306 SP BR,SELB,SPO
1447 006310 BRWTE MEMX,AORB,SPO,INCMAR
1448 006312 OUT BR,SELB,LOFICO,INCMAR ; OUTPUT CTRL
1449 006314 OUT MEMX,SELB,LOFIDA ; OUTPUT DATA
1450 006316 BP4 LU20 ; FIFOE, EXIT
1451 006320 ALWAYS GAMES ; END MESSAGE
1452 ;
1453 ;
    
```

\*KMC\* LOAD BASE ADDRESS

```

1455          .SBTTL *KMC*  LOAD BASE ADDRESS
1456          ;
1457          ; *****
1458          ; * LOAD BASE ADDRESS *
1459          ; *****
1460          ;
1461          ;
1462          ; THE THREE BYTES OF BASE ADDRESS
1463          ; ARE SAVED IN MAR
1464          ; THE CONTENTS OF BASE ADDRESS IS
1465          ; READ TO CHECK IF MEMORY EXISTENT
1466          ;
1467          ;
1468 006322     PA00:  LDMA  IMM,2                ;BAD ADDRESS
1469          ;
1470 006324     LDMA  IMM,BADA
1471 006326     MEM   IBUS,PORT1                ; LOW
1472 006330     OUT   MEMX,SELB,IBA1,INCMAR
1473 006332     MEM   IBUS,PORT2                ; MEDIUM
1474 006334     OUT   MEMX,SELB,IBA2,INCMAR
1475 006336     SP    IBUS,PORT14,SP8           ; HIGH
1476 006340     CALLSB SP5,HIPA10
1477 006346     MEM   IMM,0
1478 006350     SPBR  BR,INCA,SP8
1479          ;
1480 006352     OUT   BR,SELB,ONPR                ; RO DMA
1481 006354     PA001: BRWRTE  IRUS,NPR
1482 006356     BRU   PA001
1483 006360     BRWRTE  IBUS,UBBR                ; EXISTENT
1484 006362     BRO   NOBAD                       ; MEMORY?
1485          ;
1486          ;
1487 006364     MEM   MEMX,SELA,SP8
1488 006366     ALWAYS OKMESS                    ; OK MESSAGE
1489          ;

```

```

1491 .SBTTL *KMC* LOAD MASK
1492 ;
1493 ; *****
1494 ; * LOAD MASK *
1495 ; *****
1496 ;
1497 ; READ STATUS HIGH
1498 ; IF L.U. OFF LINE, ERROR MESSAGE AND EXIT
1499 ;
1500 ; LOAD AND MASK (BSEL5) IN STATUS LOW
1501 ; LOAD COMMAND IN STATUS HIGH
1502 ;
1503 ; WAIT UNTIL L.U. ANSWERS IN STATUS HIGH
1504 ; LOAD OR MASK (BSEL4) IN STATUS LOW
1505 ; LOAD COMMAND IN STATUS HIGH
1506 ; EXIT
1507 ;
1508 ;
1509 ;
1510 006370 PE00: CALLSE SP5,IXINI ; TEST IF ON LINE
1511 ;
1512 006376 OUT IEUS,PORT12,LOSTLO ; SEND AND MASK
1513 006400 BRWRTE IMM,AMASK
1514 006402 OUT BR,SELB,LOSTHI
1515 006404 PE10: BRWRTE IEUS,RESCAN ; WAIT ANSWER
1516 006406 BR4 PE20
1517 006410 ALWAYS PE10
1518 ;
1519 006412 PE20: NOLST IRUS,RESTHI ;ASSUME ACK
1520 006414 OUT IBUS,PORT1,LOSTLO ; SEND OR MASK
1521 006416 BRWRTE IMM,XMASK
1522 006420 OUT BR,SELB,LOSTHI
1523 ;
1524 006422 ALWAYS SC30 ; EXIT
1525 ;

```

```
1527 .SBTTL *KMC* APORT TRANSMISSION
1528 ;
1529 ; *****
1530 ; * APORT TRANSMISSION *
1531 ; *****
1532 ;
1533 ; THIS ROUTINE IS ENTERED WHEN AN APORT IS REQUESTED
1534 ; BY THE PDP
1535 ;
1536 ; CHECK IF A TRANSMISSION IS GOING ON
1537 ; IF NOT, EXIT
1538 ; NEW STATE IS APORT
1539 ; EXIT
1540 ;
1541 ;
1542 ;
1543 006424 PFA00: SP IMM,0,SPU,LDMAPG ; PAGE 0
1544 006426 NODST MEMX,SUB,SP0 ; GET TX STATE
1545 006430 C SC30 ; FREE, EXIT
1546 ;
1547 006432 MEM IMM,IXABOR ; NEW STATE
1548 006434 ALWAYS SC30 ; EXIT
1549 ;
1550 ;
1551 ;
1552 ; *****
1553 ; * APORT TRANSMISSION AFTER REQUEST *
1554 ; *****
1555 ;
1556 ;
1557 ; SEND APORT TO FIFO CONTROL
1558 ; SEND ANYTHING TO FIFO DATA TO PULL THE QUEUE
1559 ; SEND ERROR MESSAGE TO PDP
1560 ;
1561 ;
1562 ;
1563 006436 PFA00: BRWRT IMM,APORT
1564 006440 ALWAYS PFB10
```

```
1566 ;
1567 ;
1568 ;
1569 ;
1570 ;
1571 ; *****
1572 ; * ABORT TRANSMISSION AFTER NON-EXISTENT MEMORY *
1573 ; *****
1574 ;
1575 ; SEND ABORT TO FIFO CONTROL
1576 ; SEND ANYTHING TO FIFO DATA TO PULL THE QUEUE
1577 ; SEND ERROR MESSAGE TO PDP
1578 ;
1579 ;
1580 ;
1581 006442 PFB00: BPWRITE IMM,NOEXME
1582 006444 PFB10: MEM IMM,<EOF!BOF> ; OUTPUT ABORT
1583 006446 OUT MEMX,SELB,LOFICO
1584 006450 OUT BR,SELB,LOFIDA
1585 006452 NODST BR,SELA,SP1,LDMAR ; NEW STATE = FREE
1586 006454 MEM IMM,KFREE
1587 ;
1588 006456 ALWAYS ERKACA ; ERROR MESSAGE
1589 ;
1590 ;
```

```

1592          .SBTTL *KMC*  ABORT RECEPTION
1593          ;
1594          ; *****
1595          ; * APOPI RECEPTION *
1596          ; *****
1597          ;
1598          ; THIS ROUTINE IS ENTERED WHEN REQUESTED BY CUSTOMER
1599          ; EXIT IF DUMMY COMMAND
1600          ; ELSE,
1601          ; CHECK IF STATUS IS FREE
1602          ; IF NOT, NEW STATE IS ABORT REQUESTED
1603          ; CLEAR TAB ADDRESS FLAG
1604          ; QUEUE ERROR MESSAGE IN ALL CASES
1605          ;
1606          ;
1607          ;
1608          ;
1609          ;
1610 006460     PG00: LDMAP   IMM,1
1611 006462     BRWRT  IMM,20          ; TEST IF DUMMY COMMAND
1612 006464     COMP   BR,SP3
1613 006466     Z      SC00          ; EXIT IF DUMMY
1614          ;
1615 006470     SP     IMM,0,SP0
1616 006472     COMP   MEMX,SP0          ; TEST IF FREE
1617 006474     Z      PG10
1618 006476     MEM    IMM,ARORR          ; NEW STATE = ABORT REQUESTED
1619 006500     PG10: BRWRT  IMM,BAROLD,INCMAR ; ERROR = ABORT REQUESTED
1620 006502     INCMR
1621 006504     INCMR
1622 006506     MEM    IMM,0          ; CLEAR TAB ADDRESS FLAG
1623 006510     ALWAYS ERKMC          ; SEND ERROR
1624          ;
1625          ;
1626          ;
1627          ;

```

1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640 006512  
1641 006514  
1642 006516  
1643 006520  
1644 006522  
1645  
1646  
1647

```
.SBTTL *KMC* MISCELLANEOUS COMMANDS  
;  
; *****  
; * MISCELLANEOUS COMMANDS *  
; *****  
;  
; TEST IF UNIT ON LINE  
; SEND COMMAND IN STATUS HIGH  
;  
;  
PH00: BRWTE IBUS,RELOAD ; TEST IF ON LINE  
BR4 PH01 ; ON LINE  
ALWAYS TXIN11 ; OFF LINE, ERROR MESSAGE  
PH01: OUT IBUS,PORT3,LOSTHI  
ALWAYS SC30  
;  
;  
;
```



```

1649 .SR11L *KMC* RECEPTION ROUTINES
1650 ;
1651 ; *****
1652 ; * FREE *
1653 ; *****
1654 ;
1655 ;
1656 ; SET OVERFALL COUNT TO ZERO
1657 ; NEW STATE = WT FOR CONTROL CHARACTER
1658 ; SAVE FIFO DATA IN MAR + 7
1659 ; GO TO TEST FIFO
1660 ;
1661 ;
1662 006524 FI0000: BRWRTI IMM,4
1663 006526 NODST BR,ADD,SP1,LDMAR
1664 ;
1665 ;
1666 ;
1667 ;
1668 ;
1669 ;
1670 ;
1671 ;
1672 ;
1673 ;
1674 ;
1675 ;
1676 ;
1677 ;
1678 ;
1679 ;
1680 006530 3s:
1681 006530 MEM IMM,0,INCMAR ; INIT COUNT
1682 006532 MEM IMM,0
1683 ;
1684 006534 SP MEMX,SELB,SP14
1685 006536 SP MEMX,SELB,SP15,INCMAR
1686 ;
1687 006540 BRWRTI IMM,7 ; SAVE TYPE OUT
1688 006542 BRWRTI BR,AANDB,SP11
1689 006544 BRODAT
1690 006546 MEM BR,SELB,INCMAR ; FLAG FOR TAB ADDRESS
1691 006550 SP BR,SELB,SP2 ; IS IN BIT 7
1692 ;
1693 006552 BRWRTI IMM,WTCTRL ; NEW STATE
1694 ;
1698 006554 MEM IBUS,REFIDA
1700 ;
1701 006556 ALWAYS LUF157
1702 ;
1703 ;
1704 ;
  
```

```
1706 ;  
1707 ; *****  
1708 ; * BEGIN OF FRAME AND END OF FRAME ? *  
1709 ; *****  
1710 ;  
1711 ; READ CHARACTER  
1712 ; EXIT  
1713 ;  
1714 ;  
1715 ;  
1716 006560 FI0100: BRWETE IBUS,PEFIDA  
1717 006562 ALWAYS LU10  
1718 ;  
1719 ;  
1720 ;
```

```
1722 ;  
1723 ; *****  
1724 ; * WAITING FOR CONTROL CHARACTER *  
1725 ; *****  
1726 ;  
1727 ; READ AND SAVE FIFO DATA  
1728 ; LOAD NEW STATE DEPENDING UPON TYPE  
1729 ; IF LONG CONTROL FRAME : STATE : DATA 1  
1730 ; IF INFORMATION FRAME : STATE : INFO FRAME, NO BUFFER  
1731 ; BRANCH TO TEST IF FIFO READY  
1732 ;  
1733 ;  
1734 ;  
1735 006564 FI0200: BRWRT IMM,10  
1736 006566 NODST BP,ADD,SP1,LDMAK  
1737 ;  
1738 006570 MFM IPUS,REFIDA ; SAVE DATA IN MAR  
1739 006572 BRWRT BP,SELA,SP2  
1740 006574 BR1 FI0210  
1741 006576 BPWRT IMM,INOB ; INFO FRAME  
1742 006600 ALWAYS LUF157  
1743 006602 FI0210: BRWRT IMM,LFD1 ; CONTROL FRAME  
1744 006604 ALWAYS LUF157  
1745 ;  
1746 ;  
1747 ;  
1748 ;
```

```
1750 ;  
1751 ; *****  
1752 ; * CONTROL CHARACTER + EOF *  
1753 ; *****  
1754 ;  
1755 ; STORE DATA IN MAR + 8  
1756 ; SEND MESSAGE UK AND EXIT  
1757 ;  
1758 ;  
1759 ;  
1760 ;  
1761 006606 F10300: BRWRT  IMM,10  
1762 006610 NODST  BP,ADD,SF1,LDMAR  
1763 ;  
1764 006612 MEM    IBUS,REFIDA           ; SAVE DATA  
1765 ;  
1766 006614 ALWAYS OKMESR  
1767 ;  
1768 ;  
1769 ;
```

```
1771 ;  
1772 ; *****  
1773 ; * LONG FRAME, DATA1 *  
1774 ; *****  
1775 ;  
1776 ; SAVE DATA IN MAR + 4  
1777 ; NEW STATE = LONG FRAME, DATA 2  
1778 ; GO TO TEST IF FIFO READY  
1779 ;  
1780 ;  
1781 ;  
1782 006616 FI0400: BRWRTI IMM,4  
1783 006620 NODST BR,ADD,SP1,LDMAR  
1784 006622 MEM IBUS,REFIDA ; SAVE DATA  
1785 006624 BRWRTI IMM,LFD2 ; NEW STATE  
1786 006626 ALWAYS LUFIS7  
1787 ;  
1788 ;  
1789 ;
```

```
1791 ;  
1792 ; *****  
1793 ; * LONG CONTROL FRAME, DATA 1 OR 2, AND EOF *  
1794 ; *****  
1795 ;  
1796 ;IT IS AN ERROR CASE  
1797 ;  
1798 ; FORCE FCS ERROR IN MAR + 14 ( ERROR STATUS )  
1799 ; BRANCH TO ERROR ROUTINE  
1800 ;  
1801 ;  
1802 ;  
1803 006630 FI0500: BRWRITE IMM,FCSERR  
1804 006632 OUT BR,SELB,UBA1  
1805 006634 ALWAYS FI1100  
1806 ;  
1807 ;  
1808 ;
```

```
1810 ;  
1811 ; *****  
1812 ; * LONG CONTROL FRAME, DATA 2 *  
1813 ; *****  
1814 ;  
1815 ;SAVE DATA IN MAR + 5  
1816 ; NEW STATE = LONG FRAME , DATA 3  
1817 ; GO TO TEST IF FIFO READY  
1818 ;  
1819 ;  
1820 ;  
1821 ;  
1822 006636 FI0600: BRWTE IMM,5  
1823 006640 NQDST FF,ADD,SP1,LDMAR  
1824 006642 MEM IBUS,REFIDA ; SAVE DATA 2  
1825 006644 BPWTE IMM,LFD3 ; NEW STATE  
1826 006646 ALWAYS LUF157  
1827 ;  
1828 ;  
1829 ;
```

```
1831 ;  
1832 ; *****  
1833 ; * LONG CONTROL FRAME, DATA 3 ; ERROR CASE *  
1834 ; *****  
1835 ;  
1836 ; IT IS AN ERROR CASE  
1837 ;  
1838 ; NEW STATE = APORT RX  
1839 ; READ FIFO DATA  
1840 ; FORCE FCS ERROR IN ERROR STATUS ( MAR + 14)  
1841 ;  
1842 ;  
1843 ;  
1844 006650 FI0800: BRWRTE IMM,ABOR ; NEW STATE  
1845 006652 SP BR,SELB,SP10  
1846 006654 BRWRTE IMM,FCSERR  
1847 006656 OUT BR,SELB,UBA1 ; FORCE FCS ERROR  
1848 006660 ALWAYS F11000  
1849 ;  
1850 ;  
1851 ;
```



```
1853 ;  
1854 ; *****  
1855 ; * LONG CONTROL FRAME, DATA 3 AND END OF FRAME*  
1856 ; *****  
1857 ;  
1858 ; READ DATA IN SP11  
1859 ; SEND MESSAGE OK AND EXIT  
1860 ;  
1861 ;  
1862 ;  
1863 006662 F10900: SP IHUS,REFIDA,SP11  
1864 006664 ALWAYS OKMESP  
1865 ;  
1866 ;  
1867 ;
```

```
1869 ;  
1870 ; *****  
1871 ; * ABOUT RECEPTION *  
1872 ; *****  
1873 ;  
1874 ;  
1875 ;  
1876 ;  
1877 ; READ FIFO DATA  
1878 ; GO TO TEST IF FIFO FREE  
1879 ;  
1880 ;  
1881 ;  
1882 006666 FI1000: BRWTF 1BUS,REF1DA ; READ DATA  
1883 006670 ALWAYS LUF162  
1884 ;  
1885 ;  
1886 ;
```

```

1888 ;
1889 ; *****
1890 ; * ABORT RECEPTION AND END OF FRAME *
1891 ; *****
1892 ;
1893 ; AN ERROR MESSAGE HAS TO BE SEND
1894 ;
1895 ; READ LAST FIFO DATA
1896 ; THE ERROR STATUSES ARE IN NPR6 AND MAR + 15
1897 ; CALL GENERAL ERROR ROUTINE
1898 ;
1899 ;
1900 ;
1901 006672 FI1100: NODST ISUS,PEFIDA ; READ DATA
1902 006674 FI1110: BRWFE 1MM,17
1903 006676 NODST BR,ADD,SP1,LDMAR
1904 006700 FI1120: OUT MEMX,SELB,UBA2 ; ERROR KMC
1905 006702 MEM 1MM,0
1906 006704 OUT MEMX,SELB,UBA2 ; NO TX ERROR
1907 006706 NODST BR,SELA,SP1,LDMAR
1908 006710 MEM 1MM,RXFREE ; NEW STATE
1909 006712 ALWAYS ERGEN
1910 ;
1911 ;
1912 ;
    
```

```

1914 ;
1915 ; *****
1916 ; * INFO FRAME, NO BUFFER AVAILABLE *
1917 ; *****
1918 ;
1919 ; CHECK IF THERE IS THE NEW BUFFER FLAG IN MAR + 3
1920 ; IF NO, EXIT
1921 ;
1922 ; CHECK IF IT IS THE LAST BUFFER
1923 ; IF YES, SET A RECEIVER BUFFER OVERFLOW ERROR
1924 ; AND NEW STATE IS ABORT RX
1925 ;
1926 ; SAVE CHARACTER ADDRESS
1927 ; IN NPK6 ( ADDRESS LOW )
1928 ; IN NPK7 ( ADDRESS MEDIUM )
1929 ; IN MISC ( ADDRESS HIGH )
1930 ;
1931 ; NEW STATE = IN1E OR IN2E, DEPENDING UPON PARITY OF ADDRESS LOW
1932 ; SAVE COUNT IN SP14 AND SP15
1933 ; ADD COUNT TO OVERALL COUNT ( MAR + 4 AND + 5 )
1934 ; FLAG LAST BUFFER IN MAR + 3 IF NECESSARY
1935 ;
1936 ;
1937 ;
1938 ; 1 - TEST NEW BUFFER FLAG
1939 ;
1940 006714 FI1200: BRWRTI IMM,3
1941 006716 NODST BR,ADD,SP1,LDMAR
1942 ;
1943 006720 SPBF MEMX,SELB,SP0 ; READ FLAG
1944 006722 BR0 FI1221
1945 006724 NODST BR,SELA,SP1,LDMAR ; RESET STATE
1946 006726 MEM IMM,INOB
1947 006730 ALWAYS LUI0 ; NOT YET FLAGGED
1948 ;
1949 ;
1950 ;
1951 ; 2 - TEST LAST BUFFER FLAG
1952 ;
1953 006732 FI1220: BRWRTI IMM,3
1954 006734 NODST BR,ADD,SP1,LDMAR
1955 ;
1956 006736 BRWRTI MEMX,SELB ; READ FLAG
1957 006740 FI1221: BR7 FI1250 ; ERROR
1958 006742 BRADDE FI1540 ; PREPARE EXIT ADDRESS
1959 006744 SP BR,SELB,SP11 ; IF NON EX MEM
1960 ;
1961 ;
1962 ; 3 - FIRST DMA TO GET CHARACTER ADDRESS
1963 ;
1964 006746 CALLSF SP8,TABAD,TABAR2 ; DMA
1965 006754 FI1225: OUF IPUS,INDAT1,OBA1 ; SAVE DATA
1966 006756 OUF IPUS,INDAT2,OBA2
1967 006760 BRWRTI IBUS,INDAT1 ; CHECK PARITY
1968 006762 BR0 FI1230
1969 006764 BRWRTI IMM,IN1F ; EVEN ADDRESS
1970 006766 ALWAYS FI1240
    
```

```

1971 006770
1972 006772
1973
1974
1975
1976
1977
1978 006774
1979 007002
1980 007004
1981 007006
1982 007010
1983 007012
1984 007014
1985 007016
1986
1987
1988
1989
1990
1991 007020
1992
1993
1994
1995
1996
1997
1998
1999 007022
2000
2004
2005 007024
2006 007026
2007 007030
2008 007032
2009 007034
2010 007036
2011
2012
2013
2014
2015 007040
2016 007042
2017 007044
2018 007046
2019
2020
2021

FI1230: BRWRTF IMM,1M10
FI1240: SP BR,SELB,SP10
;
;
; 4 - SECOND DMA TO GET CHARACTER COUNT
;
FI1245: CALLSR SPS,DMAC,DMACR
BRWRTF IMM,100
SP IPUS,UBBR,SPO
BRWRTF BR,AANDB,SPO
BRWRTF BR,AGRE,SP8
OUT BF,SELB,OPR
BRWRTF BR,SELA,SP3
BF7 FI1260
;
;
;
; 5 - NOW, IT IS NORMAL RECEPTION
;
ALWAYS LUF161
;
;
; 6 - SPECIAL CASES
;
; -- LAST BUFFER,ERROR
;
FI1250: BRWRTF IMM,17
;
;
NODST BR,ADD,SP1,LDMAR
MEM IMM,BUFOVE
FI1255: BRWRTF IMM,0
OUT BR,SELB,OPR1
BRWRTF IMM,ABOR
ALWAYS LUF157
;
;
; -- LAST BUFFER FLAG RECEIVED
;
FI1260: BRWRTF IMM,3
NODST BR,ADD,SP1,LDMAR
MEM MEMX,SELA,SP3
ALWAYS LUF161
;
;
;
;
; ODD ADDRESS
; ADDRESS HIGH
; MASK VECT XX4
; LAST BUFFER
; KMC ERROR BUFFER
; BUFFER OVERFLOW
; NO RX ERROR
; NEW STATE
; EXIT
; SET FLAG IN MAR
; EXIT
    
```

```
2023 ;  
2024 ;  
2025 ; *****  
2026 ; * ABORT REQUESTED BY PDF AND END-OF-FRAME *  
2027 ; *****  
2028 ;  
2029 ;  
2030 ; READ DATA  
2031 ; NEW SIATE = FREE  
2032 ;  
2033 ;  
2034 ;  
2035 007050 FI1300: WODST IPUS,REFIDA  
2036 007052 WODST BR,SELA,SP1,LDMAH  
2037 007054 MEM IMM,RXFREE  
2038 007056 ALWAYS LU10  
2039 ;  
2040 ;  
2041 ;
```

```
2043 ;  
2044 ; *****  
2045 ; * INFO FRAME, NEW BUFFER, EVEN ADDRESS *  
2046 ; *****  
2047 ;  
2048 ; SAVE DATA IN NPR1  
2049 ; NEW STATE = NEW BUFFER, SECOND CHARACTER  
2050 ;  
2051 ;  
2052 ;  
2053 007060 FI1400: OUT IHUS,REFIDA,OIDAT2  
2054 007062 BRWPTF IMM,IN2E  
2055 007064 ALWAYS LUF160  
2056 ;  
2057 ;  
2058 ;
```

\*KMC\* RECEPTION ROUTINES

```

2060 ;
2061 ; *****
2062 ; * INFO FRAME, NEW BUFFER, EVEN ADDRESS, END OF FRAME *
2063 ; *****
2064 ;
2065 ;
2066 ;
2067 ;
2068 007066 FI1500: OUT IBUS,REFIDA,OUTDA1
2069 007070 FI1510: BRWRTI IMM,221 ; DMA 1 BYTE
2070 007072 FI1520: OUT BR,SELB,ONPR
2071 007074 FI1530: BRWRTI IBUS,NPR ; WAIT END OF DMA
2072 007076 BRU FI1530
2073 007100 BRWRTI IBUS,UBBR ; NON EX MEM
2074 007102 BPO FI1540 ; END OK, MESSAGE, EXIT
2075 007104 ALWAYS OKMESP
2076 ;
2077 007106 FI1540: BRWRTI IMM,17 ; NON EX MEM
2078 007110 NODST BR,ADD,SP1,LDMAR
2079 007112 MEM IMM,NOEXME
2080 007114 BRWRTI IMM,100
2081 007116 SP IBUS,UBBR,SP0 ; MASK VECT XX4
2082 007120 OUT BR,AANDE,OBR ; CLEAR ERROR FLAG
2083 007122 ALWAYS FI1255 ; MESSAGE AND EXIT
2084 ;
2085 ;
2086 ;

```



2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103 007124  
2104 007126  
2105 007130  
2106 007132  
2107 007134  
2108 007136  
2109 007140  
2110 007142  
2111 007144  
2112  
2113  
2114

```
;  
; *****  
; * INFC FRAME, NEW BUFFER, ODD ADDRESS *  
; *****  
;  
; LOAD DATA IN NPR3  
; INITIATE A DMA FOR 1 BYTE  
; WAIT END OF DMA  
; MAKE CHARACTER ADDRESS EVEN  
; NEW STATE = BODY OF BUFFER, EVEN ADDRESS, 1ST CHARACTER  
; BRANCH TO DISPATCHING ROUTINE  
;  
;  
;  
FI1600: OUT 1BUS,REFIDA,OUTDA2  
BRWRTI IMM,221  
OUT BR,SELB,UNPR ; START DMA  
FI1610: BRWRTI 1BUS,NPR  
BR0 FI1610  
SP 1BUS,IOBA1,SPO ; MAKE ADDRESS EVEN  
OUT BR,DECA,DBA1  
BRWRTI IMM,1R1 ; NEW STATE  
ALWAYS LUF160  
;  
;  
;
```

```
2116 ;  
2117 ; *****  
2118 ; * INFO FRAME, NEW BUFFER, ODD ADDRESS, END OF FRAME *  
2119 ; *****  
2120 ;  
2121 ; LOAD DATA IN NPR3  
2122 ; BRANCH TO THE PART THAT INITIATES A DMA FOR 1 BYTE,  
2123 ; THEN SENDS A MESSAGE OK  
2124 ;  
2125 ;  
2126 ;  
2127 007146 FI1700: OUI IEUS,REFIDA,OUTDA2  
2128 007150 ALWAYS FI1510  
2129 ;  
2130 ;  
2131 ;
```

2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146

```
;  
; *****  
; * INFO FRAME, NEW BUFFER, 2ND CHARACTER *  
; *****  
;  
; BRANCH TO ADDRESS ( F12210 ) WHERE FIFO DATA IS LOADED IN NPR3  
; SAVED DATA IS LOADED IN NPR2,  
; DMA FOR 2 BYTES IS INITIATED  
; NEW STATE IS BODY OF BUFFER, EVEN ADDRESS, 1ST CHARACTER  
;  
;  
;  
;
```

```
2148 ;  
2149 ; *****  
2150 ; * INFO FRAME, NEW BUFFER, 2ND CHARACTER, END OF FRAME *  
2151 ; *****  
2152 ;  
2153 ; LOAD FIFO DATA IN NPR3  
2154 ; LOAD SAVED DATA IN NPR2  
2155 ; INITIATE 2 BYTE DMA  
2156 ; BRANCH TO PIECE OF CODE THAT SENDS A MESSAGE OK  
2157 ;  
2158 ;  
2159 ;  
2160 007152 F11900: OUT IHS,INDAT2,OUTDA1 ; PREPARE DATA  
2161 007154 OUT IRUS,REFIDA,OUTDA2  
2162 007156 BRWRT: IMM,21 ; PREPARE DMA  
2163 007160 ALWAYS F11520  
2164 ;  
2165 ;  
2166 ;
```

```
2168 ;  
2169 ; *****  
2170 ; * INFO FRAME, BODY OF BUFFER, 1ST CHARACTER *  
2171 ; *****  
2172 ;  
2173 ; SAVE DATA IN NPR1  
2174 ; NEW STATE = BODY OF BUFFER, 2ND CHARACTER  
2175 ;  
2176 ; THIS IS THE FIRST MOST COMMON CASE  
2177 ;  
2178 ;  
2179 ;  
2180 007162 FI2000: OUT IPUS,REFIDA,OIDAT2 ; SAVE DATA  
2181 007164 BRWRTE IMM,IB2 ; NEW STATE  
2182 007166 ALWAYS LUF160  
2183 ;  
2184 ;  
2185 ;
```

2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201 007170  
2202  
2203  
2204

```
;  
; *****  
; * INFO FRAME, BODY OF BUFFER, 1ST CHARACTER, END OF FRAME *  
; *****  
;  
; WAIT FOR LAST DMA  
; ADD 2 TO CHARACTER ADDRESS  
; BRANCH TO POINT WHERE  
; DATA IS LOADED IN NPR2  
; DMA IS INITIATED FOR 1 BYTE  
; EXIT TO SEND MESSAGE OK  
;  
;  
;  
FI2100: CALLSR SP5,RXDMA0,FI1500  
;  
;  
;
```

```

2206 ;
2207 ; *****
2208 ; * INFO FRAME, BODY OF BUFFER, 2ND CHARAC *
2209 ; *****
2210 ;
2211 ; WAIT UNTIL LAST DMA OVER
2212 ;CHECK IF NON EXISTENT MEMORY
2213 ; IF NON EXISTENT, BRANCH TO ERROR MESSAGE AND EXIT
2214 ; INCREMENT ADDRESS BY 2
2215 ; LOAD FIFO DATA IN NPR3
2216 ; LOAD SAVED DATA IN NPR2
2217 ; INITIATE A 2 BYTE DMA
2218 ; NEW STATE = BODY OF FRAME, 1ST CHARACTER
2219 ;
2220 ; THIS IS THE SECOND MOST COMMON ROUTINE
2221 ;
2222 ;
2223 ;
2224 007176 FI2200: BRWRTE 1BUS,NPR ; WAIT END OF DMA
2225 007200 BRO FI2200
2226 007202 BRWRTE 1BUS,UBRR
2227 007204 BRO F11540 ; NON EX MEM
2228 007206 SP 1BUS,IOBA1,SP0 ; INC ADDRESS LOW
2229 007210 SP ER,INCA,SP0
2230 007212 OUT BR,INCA,OBA1
2231 007214 C FI2220
2232 ;
2233 007216 FI2210: OUT 1BUS,INDAT2,OUTDA1 ; PREPARE DATA
2234 007220 OUT 1BUS,REFIDA,OUTDA2
2235 007222 BRWRTE 1MN,21
2236 007224 OUT BR,SELB,ONPR ; DMA
2237 007226 BRWRTE 1MN,1B1 ; NEW STATE
2238 007230 ALWAYS LUF160
2239 ;
2240 ;
2241 007232 FI2220: CALLSR SP5,RXDMA2,FI2210 ; INC ADDRESS MED AND HIGH
2242 ;
2243 ;

```

```
2245 ;  
2246 ; *****  
2247 ; * INFO FRAME, BODY OF BUFFER, 2ND CHARACTER, EOF *  
2248 ; *****  
2249 ;  
2250 ; WAIT UNTIL LAST DMA PERFORMED  
2251 ; CHECK IF MEMORY INEXISTENT  
2252 ; IF YFS, BRANCH TO ERROR MESSAGE AND EXIT  
2253 ; INCREMENT ADDRESS BY 2  
2254 ; BRANCH TO ROUTINE THAT SENDS DMA AND MESSAGGE OK TO PDP  
2255 ;  
2256 ;  
2257 ;  
2258 ;  
2259 007240 FI2300: CALLSR SP5,RXDMA0,FI1900  
2260 ;  
2261 ;
```



```
2263 ;  
2264 ; *****  
2265 ; * INFO FRAME, END OF BUFFER, FIRST CHARACTER *  
2266 ; *****  
2267 ;  
2268 ; WAIT END OF DMA  
2269 ; INITIATE DMA TO SEND SAVED DATA  
2270 ; BRANCH TO ROUTINE THAT GIVES A NEW BUFFER  
2271 ;  
2272 ;  
2273 ;  
2274 ;  
2275 007246 FI2400: CALLSR SF5,KXDMA0 ; WAIT DMA  
2276 ; AND INC ADDRESS  
2277 007254 OUT IBUS,INDAT2,OUTDA1 ; PREPARE DATA  
2278 007256 BRWRT IMM,221  
2279 007260 OUT BR,SELB,ONPR ; REQUEST DMA  
2280 007262 FI2410: BRWRT IBUS,NPR ; WAIT DMA  
2281 007264 BR0 FI2410  
2282 007266 BRWRT IBUS,UBBR  
2283 007270 BR0 FI1540 ; NON EX MEM  
2284 007272 ALWAYS FI1220  
2285 ;  
2286 ;  
2287 ;
```

2289  
2290  
2291  
2292  
2293  
2294 007274  
2295  
2296  
2297

.SBITL \*KMC\* RELAY ADDRESS IN PAGE 2  
;  
; SERVICING OF OUTPUT TO PDP  
; ROUTINE THAT INCREMENT NPR ADDRESS BY 2  
;  
SCREL: ALWAYS SC411  
;  
;  
;

```

2299 .SBTTL *KMC* FIFO OUTPUT READY
2300 ;
2301 ; *****
2302 ; * FIFO OUTPUT READY AND TRANSMISSION *
2303 ; *****
2304 ;
2305 ; THIS ROUTINE IS ENTERED WHEN :
2306 ; FIFO OUTPUT IS READY
2307 ; AND TRANSMISSION IS GOING ON
2308 ;
2309 ; PAGE 0
2310 ; THE CONTEXT IS LOADED, THAT IS :
2311 ; TX STATE
2312 ; DATA SAVED
2313 ; COUNT
2314 ; CHARACTER ADDRESS
2315 ; BRANCH TO ROUTINE DEPENDING UPON TX STATE
2316 ;
2317 ;
2318 ;
2319 007276 LUFO0: SP BR,AANDB,SP6 ; UPDATE SCAN
2320 007300 LDMAP IMM,0 ; PAGE 0
2321 007302 BRWRTE IMM,6
2322 007304 NODST BR,ADD,SP1,LDMAR
2323 007306 SP MEMX,SELB,SP2 ; TYPE OUT
2324 007310 WODST BR,SELA,SP1,LDMAR
2325 007312 SP MEMX,SELB,SP10,INCMAR ; LOAD STATUS
2326 ;
2327 007314 LUFO10: BRADDR LUFO20 ; DISPATCH
2328 007316 .ALWAY BR,ADD,SP10,P2
2329 007320 LUFO20: ALWAYS LU20 ; FREE ?
2330 007322 ALWAYS LU20 ; FREE ?
2331 007324 ALWAYS PBA00 ; BEGIN OF FRAME
2332 007326 ALWAYS LUFO30 ; BEGIN OF BUFFER ODD
2333 007330 ALWAYS LUFO40 ; SECOND CHARACTER
2334 007332 ALWAYS LUFO30 ; FIRST CHARACTER
2335 007334 ALWAYS PDA00 ; SHORT FRAME : ADDRESS
2336 007336 ALWAYS PDB00 ; SHORT FRAME : CONTROL
2337 007340 ALWAYS PCD00 ; LONG FRAME: DATA 3
2338 007342 INCMAR ; DATA 2
2339 007344 ALWAYS PCC00 ; DATA 1
2340 007346 ALWAYS PCB00 ; CONTROL
2341 007350 ALWAYS PCA00 ; ADDRESS
2342 007352 ALWAYS PFA00 ; TX ABORT
2343 007354 ALWAYS PFB00 ; NON EX MEM
2344 ;
2345 ;
2346 ;
2347 007356 LUFO30: CALLSP SP5,LOCONT,PBC00 ; LOAD CONTEXT
2348 007364 LUFO40: CALLSR SP5,LOCONT,PBD00
2349 ;
2350 ;
2351 ;
    
```

\*KMC\* ERROR STATUS

```

2353 .SBTTL *KMC* ERROR STATUS
2354 ;
2355 ; *****
2356 ; * ERROR DETECTED BY LINE UNIT *
2357 ; *****
2358 ;
2359 ; IT MAY BE AN ERROR RELATED TO TRANSMISSION
2360 ; OR TO COMMAND OR TO BOTH
2361 ;
2362 ; THE ERROR IS IN SP12
2363 ; LOAD 0 IN NPR6 AND 7 TO SAY THERE IS NO OTHER ERROR
2364 ; LOOK IF IT IS A TRANSMISSION ERROR
2365 ; IF YES, PAGE 0
2366 ; TEST TX STATUS
2367 ; IF NOT FREE, LOAD TYPE OUT (MAR + 6) IN SP2
2368 ; SEND ERROR MESSAGE TO PDP
2369 ; IF NOT, LOAD TYPE OUT FOR COMMAND IN SP2
2370 ; SEND ERROR MESSAGE TO PDP
2371 ;
2372 ;
2373 ;
2374 007372 LUER0:
2375 ;
2380 007372 SPBR IMM,0,SP0,LDMAPG ; PAGE 0
2381 007374 OUT BR,SELB,0BA1 ; CLEAR RX ERROR STATUS
2382 007376 OUT BR,SELB,0BA2
2383 ;
2384 007400 BRWRTE BR,SELA,SP12
2385 007402 OUT BR,SELB,1BA2
2386 007404 BR4 LUER30 ; COMMAND ERROR
2387 ;
2388 007406 BRWRTE IMM,BEGFRA
2389 007410 LUER10: NODST BR,SELA,SP1,LDMAR ; TEST IF TX GOING ON
2390 007412 SP MEMX,SELB,SP10
2391 007414 COMP BR,SP10
2392 007416 C LUER40 ; TX
2393 ;
2394 007420 LUER30: BRWRTE IMM,COMMAN ; NO TX, DO AS IF ERROR
2395 007422 SP BR,SELB,SP2 ; TYPE OUT = COMMAND
2396 007424 ALWAYS ERGEN
2397 ;
2398 007426 LUER40: MEM IMM,KFREE ; NEW STATE = FREE
2399 007430 BRWRTE IMM,6 ; LOAD TYPE OUT
2400 007432 NODST BR,ADD,SP1,LDMAR
2401 007434 SP MEMX,SELB,SP2
2402 007436 ALWAYS ERGEN ; SEND ERROR MESSAGE
2403 ;
2404 ;

```

2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416 007440  
2417 007442  
2418 007444  
2419  
2420  
2421

```
.SBTTL *KMC* MODEM STATUS RECEIVED  
;  
; *****  
; * MODEM STATUS RECEIVED *  
; *****  
;  
; SEND MESSAGE TO PDP  
;  
;  
LUMSO: BRWRIE IMM,MODSTA ; TYPE OUT  
SP BK,SELB,SP2  
ALWAYS OKMOD  
;  
;  
;
```

2423  
2424  
2425  
2426  
2427  
2428  
2429  
2430  
2431  
2432  
2433  
2434  
2435 007446  
2436 007450  
2437 007452  
2438  
2439  
2440

```
.SETTL *KMC* COMMAND ACKNOWLEDGE  
;  
; *****  
; * COMMAND ACKNOWLEDGE *  
; *****  
;  
; PAGE 2  
; CLEAR COMMAND FLAG  
; SEND MESSAGE OK TO PDP  
;  
;  
LUACO: BR*RTB IMM,COMMAN  
SP BR,SELB,SP2  
ALWAYS UKNESS  
;  
;  
;
```

```
; TYPE OUT  
; QUEUE MESSAGE OK
```

2442  
2443  
2444  
2445  
2446  
2447  
2448  
2449  
2450  
2451  
2452  
2453  
2454 007454  
2455 007456  
2456 007460  
2457 007462  
2458 007464  
2459  
2460  
2461

```
.SBTTL *KMC*    FIFO TRACE  
;  
; *****  
; * FIFO TRACE DETECTED BY LINE UNIT *  
; *****  
;  
; PAGE 0  
; GET TYPE OUT IN MAR + 6  
; SEND MESSAGE OK TO PDP  
;  
;  
LUFTO: LDMAP    IMM,0  
       BRWRTF  IMM,6  
       MODST  BR,ADD,SP1,LDMAR  
       SP     MEMX,SELB,SP2  
       ALWAYS UKMES  
;  
; TYPE OUT  
;  
;
```

\*KMC\* QUEUING MESSAGES

```

2463 .SETTL *KMC* QUEUING MESSAGES
2464 ;
2465 ; *****
2466 ; * QUEUING MESSAGES *
2467 ; *****
2468 ;
2469 ; THIS SECTION DEALS WITH ALL OUTPUT MESSAGES TO PDP
2470 ;
2471 ;
2472 ;
2473 007466 ERKMC: OUT BR,SELB,ORA2 ; KMC ERROR MESSAGE
2474 007470 CALLSR SP5,OUTME1,ERKMC1 ; NOT COMPLETE
2475 ;
2476 ;
2477 007476 ERKMCA: OUT BR,SELB,ORA2 ; KMC ERROR MESSAGE
2478 007500 ERKMCA: CALLSR SP5,MESCOM,ERKMC1 ; COMPLETE, NO CLEAR TAB
2479 ;
2480 ;
2481 007506 NOBAD: BRWRTE IMM,200 ; NO BAD ADDRESS
2482 007510 CALLSR SP5,OUTME1,ERKMC2
2483 ;
2484 ;
2485 007516 OKMES: NODST BR,SELA,SP1,LDMAR ; TX MESSAGE OK
2486 007520 MEM IMM,KFREE
2487 007522 OKMESS: CALLSR SP5,MESCOM,MESCSR
2488 ;
2489 ;
2490 007530 OKMESR: NODST BR,SELA,SP1,LDMAR ; RX MESSAGE OK
2491 007532 MEM IMM,RXFREE
2492 007534 BRADDR MESCSR
2493 007536 SP BR,SELB,SP5
2494 007540 OKMESQ:
2495 007540 BRWRTE BR,SELA,SP2 ; CLEAR TAB ADDRESS
2496 007542 BR7 MESCO1
2497 007544 ALWAYS MESCOM
2498 ;
2499 ;
2500 007546 OKMOD: CALLSR SP5,OUTME1,OKMS ; MODEM STATUS
2501 ;
2502 ;
2503 007554 ERGEN: ; GENERAL ERROR MESS
2504 007554 BRADDR ERGE00 ; PREPARE RETURN ADDRESS
2505 007556 SP BR,SELB,SP5
2506 007560 BRWRTE IBUS,IOBA2 ; GET ERROR 2 BYTE
2507 007562 BR0 MESCOM ; NO BUFFER, NO DELETE
2508 007564 ALWAYS OKMESG ; DELETE BUFFER IF REQ
2509 ; (30 OCT 79)

```



2511  
2512  
2513  
2514  
2515  
2516  
2517  
2518  
2519  
2520  
2521  
2522  
2523  
2524  
2525  
2526 007566  
2527 007570  
2528 007572  
2529 007574  
2530 007576  
2531 007600  
2532 007602  
2533 007604  
2534 007606  
2535 007610  
2536 007612  
2537  
2538 007614  
2539 007616

```
;  
;  
;  
; *****  
; * PREPARE COMPLETE MESSAGE *  
; *****  
;  
; CLEAR TAB ADDRESS FLAG  
; POINT MAR ON OVERALL COUNT  
; LOAD OVERALL COUNT MINUS TEMPORARY CONT IN SP14, SP15  
; LOAD 1ST AND 2ND CHARACTERS IN SP9, SP10  
; QUEUE MESSAGE  
;  
;  
; MESCOM1: BRWRTE IMM,3 ; CLEAR TAB ADDRESS  
; NODST BR,ADD,SP1,LDMAR  
; MEM IMM,0  
; MESCOM: BRWRTE IMM,4  
; NODST BR,ADD,SP1,LDMAR  
; BRWRTE BR,SELA,SP14 ; LOW COUNT  
; SP MEMX,SELB,SP14  
; SP BF,SUB,SP14,INCMAR  
; BRWRTE BF,SELA,SP15 ; HIGH COUNT  
; SP MEMX,SELB,SP15,INCMAR  
; SP BR,SUBC,SP15,INCMAR  
;  
; SP MEMX,SELB,SP9,INCMAR ; 1ST CHARACTER  
; SP MEMX,SELB,SP10 ; 2ND CHARACTER
```

```
2541 ;
2542 ;
2543 ;
2544 ;
2545 ;
2546 ; *****
2547 ; * LOAD CSR MESSAGE IN QUEUE *
2548 ; *****
2549 ;
2550 ; POINT MAR ON MESSAGE QUEUE, PAGE 2
2551 ; CASE 1 : LONG CONTROL FRAME
2552 ; DATA 1 IN SP14
2553 ; DATA 2 IN SP15
2554 ; DATA 3 IN SP11
2555 ; CASE 2 : INFORMATION FRAME
2556 ; BYTE COUNT IN SP14 AND SP15
2557 ; BIT COUNT IN SP11
2558 ; LOAD SP14, 15, 11 IN MAR
2559 ;
2560 007620 OUTME1: LDMA IMM,2 ; LAST MESSAGE
2561 007622 LDMA IMM,2776 ; POINTER
2562 007624 SP MEMX,SELB,SPO
2563 007626 SP IMM,10,SP8
2564 007630 MFM MEMX,ADD,SP8 ; INC POINTER
2565 007632 NODST BR,SELA,SPO,LDMA ; POINT ON MESSAGE
2566 007634 LDMA IMM,3
2567 007636 MEM MEMX,SELA,SP14,INCMAR
2568 007640 MFM MEMX,SELA,SP15,INCMAR
2569 007642 MEM MEMX,SELA,SP11,INCMAR ; CASE 1: DATA 3
2570 007644 MEM MEMX,SELA,SP11,INCMAR ; CASE 2 : BIT COUNT
2571 ;
2572 007646 RTNSUB SP5,P3
```

```
2574 ;
2575 ;
2576 ;
2577 ; *****
2578 ; * LOAD MODEM STATUS IN QUEUE *
2579 ; *****
2580 ;
2581 ; LOAD MODEM STATUS
2582 ; BRANCH TO QUEUE CSK MESSAGE
2583 007650 OKMS: MEM 1MM,0,INCMAR
2584 007652 MEM 1MM,0,INCMAR
2585 007654 MEM 1MM,0,INCMAR
2586 007656 MEM MEMX,SELA,SP12
2587 ;
2588 007660 ALWAYS #ESCSR
```

```
2590 ;
2591 ;
2592 ;
2593 ; *****
2594 ; * LOAD KMC ERROR MESSAGE IN QUEUE *
2595 ; *****
2596 ;
2597 ; LOAD ERROR FLAG IN SP4 WITH LINE NUMBER
2598 ; ERROR IS IN SP3
2599 ; LOAD 0 OR ERROR IN MEMORY
2600 ;
2601 ; THEN GO TO QUEUE CSR MESSAGE
2602 ;
2603 007662 ERKMC1: BRWRT IMM,300
2604 ;
2605 007664 ERKMC2: SP BR,AORB,SP4
2606 007666 MEM IMM,0,INCMAR
2607 007670 MEM IBUS,IOBA2,INCMAR
2608 007672 MEM IMM,0,INCMAR
2609 007674 MEM IMM,0
2610 ;
2611 007676 ALWAYS MESCO
```

```
2613 ;
2614 ;
2615 ;
2616 ;
2617 ; *****
2618 ; * LOAD ERROR MESSAGE IN QUEUE WHEN DETECTED BY L.U. *
2619 ; *****
2620 ; SP2: TYPE DU1
2621 ; SP4: LINE NUMBER
2622 ; SP13: STATUS HIGH
2623 ; NFR5: ERROR COMING FROM STATUS LOW (ERR.1)
2624 ; NFR6: ERROR COMING FROM RX QUEUE (ERR.3)
2625 ; NFR7: ERROR DETECTED BY KMC (ERR.2)
2626 ;
2627 007700 ;
2628 007702 ERGE00: MEM IBUS,IJRA2 ; LOAD ERROR 1
2629 007704 BRWTE IMM,300 ;**** 26-OCT-79 : ERROR FLAG ONLY
2630 SP BR,AORR,SP4 ;**** 26-OCT-79
2631 ;
2632 ;*** SP IMM,210,SP8 ; INIT ERROR CLASS 4
2633 ;*** CALLSB SP5,ERMAS,MC41 ; ERROR CLASS 4
2634 ;*** CALLSB SP5,ERMAS,MC31 ; CLASS 3
2635 ;*** CALLSB SP5,ERMAS,MC21 ; CLASS 2
2636 007706 ;*** CALLSB SP5,ERMAS,MC11 ; CLASS 1
2637 INCM
2638 007710 ;
2639 007712 MEM IBUS,IORA2 ; LOAD ERROR 2
2640 BRWTE IMM,300
2641 ;*** SP BR,SELB,SP8 ; INIT ERROR CLASS 1
2642 007714 ;*** CALLSB SP5,ERMAS,MC12 ; ERROR CLASS 1
2643 INCM
2644 007716 ;
2645 007720 MEM IMM,300 ; ADD TX ON AND RX ON
2646 007722 SP IBUS,RELOAD,SP3
2647 007724 MEM MEMX,AANDB,SP3
2648 007726 SP IBUS,IOHA1,SP3
2649 MEM MEMX,AORB,SP3
2650 ;*** SP IMM,210,SP6 ; LOAD ERROR 1
2651 ;*** CALLSB SP5,ERMAS,MC43 ; INIT ERROR CLASS 4
2652 ;*** CALLSB SP5,ERMAS,MC33 ; ERROR CLASS 4
2653 007730 ;*** CALLSB SP5,ERMAS,MC23 ; ERROR CLASS 3
2654 INCM ; ERROR CLASS 2
2655 007732 ;
2656 007734 MEM IMM,0
2657 ALWAYS MESCSR
2658 ;
2659 007736 ;
2660 ;
2661 ;*** SP BR,SELB,SP3 ; LOAD MASK
2662 ;*** NODST MEMX,AORNB,SP3 ; COMPARE
2663 ;*** Z ERMAS1
2664 ;*** BRWRIE IMM,200
2665 ;*** BRWTE BR,AORR,SP8
2666 007736 ;*** SP BR,AORB,SP4 ; LOAD CLASS IN SP4
2667 ;
2668 ;*** SP BR,IWDA,SP8 ; INCREMENT CLASS
2669 ;*** RTNSUB SP5,P3
```

```

2670 ;
2671 ;
2672 ;
2673 ; *****
2674 ; * LOAD MESSAGE IN QUEUE *
2675 ; *****
2676 ;
2677 ; THIS ROUTINE UPDATES THE QUEUE
2678 ; AND LOADS MESSAGE IN MEMORY
2679 ; -TYPE OUT WITHOUT RDYO IS IN SP2
2680 ; -LINE NUMBER IS IN SP4
2681 ; THEN BRANCH TO ADDRESS LOADED IN SP5
2682 ;
2683 007736 MESC SR: LD MAP IMM,2
2684 007740 NODST BR,INCA,SP0,LDMAR
2685 ;
2686 ;
2687 007742 MEM MEMX,SELA,SP10,INCMAR
2688 007744 MEM MEMX,SELA,SP9,INCMAR
2689 007746 MEM MEMX,SELA,SP4,INCMAR
2690 007750 BRWRTE IMM,7 ; CLEAR FLAG ERROR IF ANY
2691 007752 SP BR,AANDB,SP4
2692 007754 MEM IMM,57
2693 007756 SP MEMX,AANDB,SP2
2694 007760 MEM IMM,RDYO
2695 007762 MEM MEMX,AORB,SP2
2696 ;
2697 007764 BRWRTE IMM,100 ; RESET NON EX MEMORY IF ANY
2698 007766 SP IBUS,UBBR,SP0
2699 007770 OUT BR,AANDB,OBR
2700 ;
2701 007772 ALWAYS LU10
    
```

2703  
2704  
2705  
2706  
2707  
2708  
2709  
2710

```
.SBTTL *KMC*    SUBROUTINES  
;  
;  
;  
* *****  
*    SUBROUTINES SECTIONS    *  
* *****  
;  
;
```

```

2712 ;
2713 ; *****
2714 ; * TABLE ADDRESS *
2715 ; *****
2716 ;
2717 ; THIS ROUTINE
2718 ; -LOAD MAR WITH ADDRESS OF TABLE ADDRESS
2719 ; -PREPARES DMA BY LOADING TABLE ADDRESS IN
2720 ;   NPK REGISTERS
2721 ; -ALSO INCREMENTS TABLE ADDRESS BY 2 FOR
2722 ;   NEXT STEP
2723 ; -REQUESTS DMA IN
2724 ; -WAITS FOR END OF DMA
2725 ; -CHECKS IF MEMORY INEXISTENT; IF IT IS THE
2726 ;   CASE, EXIT TO ADDRESS IN SP11
2727 ;
2728 ; RETURN ADDRESS IS IN SP8
2729 ; RETURN ADDRESS IF NON EX-MEMORY IS IN SP11
2730 ;
2731 ;
2732 007774 TABAD:  NOUPT  BR,INCA,SP1,LDMAR      ; LOAD MAR
2733 007776      SP   MEMX,SELB,SP14      ; ADDRESS LOW
2734 010000      OUT  MEMX,SELB,1BA1
2735 010002      BRWTE IMM,2
2736 010004      MEM  BR,ADD,SP14,INCMAR   ; ADD 2
2737 010006      SP   MEMX,SELB,SP15      ; ADDRESS MEDIA
2738 010010      OUT  MEMX,SELB,1BA2
2739 010012      MEM  BR,APLUSC,SP15,INCMAR
2740 010014      OUT  MEMX,SELB,ONPR      ; START DMA
2741 010016      C    TABAD2              ; INC AD HIGH
2742 ;
2743 010020 TABAD1: BRWTE  IBUS,NPR
2744 010022      BRO  TABAD1              ; WAIT DMA
2745 010024      BRWTE IBUS,UBBR
2746 010026      .BRO SELA,SP11,P1       ; MEM INEX
2747 ;
2748 010030      RTNSUB SP8,P3
2749 ;
2750 010032 TABAD2: BRWTE  IMM,4
2751 010034      SP   MEMX,SELB,SP14
2752 010036      MEM  BR,ADD,SP14
2753 010040      ALWAYS TABAD1
2754 ;
2755 ;
2756 010042 TABAR1: ALWAYS PBA15
2757 010044 TABAR2: ALWAYS FI1225
2758 ;
2759 ;

```



```

2761 ;
2762 ; *****
2763 ; * WAIT DMA AND INCREMENT NPR ADDRESS BY 2 *
2764 ; *****
2765 ;
2766 ;
2767 ;
2768 010046 RXDMA0: BPWRT IBUS,NPR ; WAIT DMA
2769 010050 BPO RYDMA0
2770 010052 BPWRT IBUS,UBHR
2771 010054 BPO I11540 ; NON EX MEM
2772 010056 RXDMA1: SP IBUS,IOA1,SPO
2773 010060 SP BR,INCA,SPO ; INC ADDRESS LOW
2774 010062 OUT BR,INCA,ORA1
2775 010064 C RXDMA2
2776 010066 RTNSUB SP5,P2
2777 ;
2778 010070 RXDMA2: SP IBUS,IOA2,SPO ; INC ADDRESS MEDIUM
2779 010072 OUT BR,INCA,ORA2
2780 010074 C RXDMA3
2781 010076 RTNSUP SP5,P2
2782 ;
2783 010100 RXDMA3: BPWRT IMM,4 ; INC ADDRESS HIGH
2784 010102 SP IBUS,UBHR,SPO
2785 010104 OUT BR,ADD,ORR
2786 010106 RTNSUB SP5,P2
2787 ;
2788 ;
2789 ;
2790 ;
    
```

```

2792 .SBTTL *KMC* TRANSMISSION SUBROUTINES
2793 ;
2794 ; *****
2795 ; * TRANSMISSION SUBROUTINES *
2796 ; *****
2797 ;
2798 ;
2799 ; *****
2800 ; * INITIALISATION *
2801 ; *****
2802 ;
2803 ;
2804 ; THIS ROUTINE CHECKS IF
2805 ; L.U. PRESENT
2806 ; L.U. NOT BUSY
2807 ; IF NOT OK, IT OUTPUTS AN ERROR MESSAGE
2808 ; TO PDF AND FX11
2809 ; ELSE, IT RETURNS TO SP5
2810 ;
2811 ;
2812 010110 TXINI: SPBK IMM,0,SP0,LDMAPG ; PAGE 0
2813 010112 SP MEMX,SELB,SP10 ; 1X STATUS
2814 010114 COMP PR,SP10
2815 010116 C TXINI2 ; BUSY
2816 010120 TXINI3: BRWRTF IBUS,RELOAD ; READ LINE UNIT ADDRESS
2817 010122 .BR4 SELA,SP5,P1 ; ON LINE
2818 010124 TXINI11: BRWRTF IMM,NOTPRE ; NOT PRESENT OR OFF LINE
2819 010126 ALWAYS ERKMC
2820 ;
2821 010130 TXINI2: BRWRTF IMM,LUBUSY
2822 010132 ALWAYS ERKMC
2823 ;
2824 ;
2825 ;
  
```

```

2827
2828 ;
2829 ; *****
2830 ; * READ PASE ADDRESS *
2831 ; *****
2832 ;
2833 ; SPO CONTAINS PAGE
2834 ; THIS ROUTINE IS CALLED AT INITIALISATION
2835 ; OF TX OF INFO FRAME
2836 ; OF TX OF LONG CONTROL FRAME
2837 ; OF RX OF INFO FRAME
2838 ;
2839 ; IT CHECKS IF BAD EXISTS, IF NOT EXIT IT DOES
2840 ; FIRST DMA AND LOAD 2 BYTES
2841 ; IN MAR (MAR IS SUPPOSED TO BE READY)
2842 ; IT DUES SECOND DMA AND LET THE BYTES
2843 ; IN NFF
2844 ;
2845 ; IT RETURNS TO ADDRESS IN SP5
2846 010134 ;
2847 010136 BADIN: LDMAP IMM,2 ; FIND BAD
2848 LDMA IMM,BADA
2849 010140 ;
2850 010142 OUT MEMX,SELB,IBA1,INCMAR
2851 010144 OUT MEMX,SELB,IBA2,INCMAR
2852 010146 BRWPIE MEMX,SELB
2853 010150 BRO BADIN3
2854 010152 ALWAYS NOBAD ; NO BAD
2855 BADIN3: OUT BR,SELB,ONPR ; RO DMA
2856 010154 ;
2857 010156 NODST MEMX,SELA,SPO,LDMAPG
2858 NODST BR,INCA,SPI,LDMAR
2859 010160 ;
2860 010162 BADINO: BRWRIE IBUS,NPF
2861 010164 BRO BADINO ; WAIT DMA
2862 010166 MEM IBUS,INDAT1,INCMAR ; STORE DATA
2863 MEM IBUS,INDAT2,INCMAR
2864 010170 ;
2865 010172 SP IBUS,IIBA1,SPO ; INC ADD LOW
2866 010174 SP BR,INCA,SPO
2867 OUT BR,INCA,IBA1
2868 010176 ;
2869 010200 SP IBUS,IIBA2,SPO ; INC ADD MED
2870 010202 OUT BR,APLUSC,IBA2
2871 010204 SP IBUS,NPR,SPO
2872 010206 C BADIN2
2873 010210 OUT BR,INCA,ONPR ; RO DMA
2874 010212 BADIN1: BRWRIE IBUS,NPF
2875 BRO BADIN1
2876 010214 ;
2877 RTNSUB SP5,P1
2878 010216 ;
2879 010220 BADIN2: BRWRIE IMM,S ; INC ADD HIGH
2880 010222 OUT BR,ADD,ONPR
2881 ALWAYS BADIN1
2882 ;
2883 ;
    
```

```

2885 ;
2886 ;
2887 ;
2888 ; *****
2889 ; * GET CHARACTER COUNT *
2890 ; *****
2891 ;
2892 ;
2893 ; THIS ROUTINE
2894 ; CALLS ROUTINE TO PERFORM DMA
2895 ; SAVES COUNT LOW AND HIGH IN SP14 AND 15
2896 ; CHECKS LAST BUFFER FLAG
2897 ; IF SET (=0), SHORTENS HIGH COUNT IN
2898 ; SP15 BY BITS AND SAVES BIT COUNT
2899 ; + LAST BUFFER FLAG IN SP3
2900 ; PREFARE NFR COMMAND WITH ADDRESS HIGH
2901 ; AND SAVE IN BRG
2902 ;
2903 ; RETURN TO ADDRESS IN SP5
2904 ;
2905 ; THIS ROUTINE IS CALLED WHEN A NEW BUFFER IS
2906 ; TO BE STARTED FOR TX OR RX
2907 ;
2908 010224 DMAC: CALLSB SP8,TABAD ; DMA
2909 010232 SP IRUS,INDAT1,SP14,INCMAR ; LOW COUN
2910 010234 MEM MEMX,ADDC,SP14,INCMAR ; OVERALL
2911 ; COUNT
2912 010236 SPBR IRUS,INDAT2,SP15 ; HIGH COUN
2913 010240 BRWRTAI
2914 010242 BR4 DMAC10
2915 ;
2916 ;
2917 010244 SP BR,SELB,SP3 ; LAST BUFFER
2918 010246 BRWRTI IMM,3
2919 010250 SP BR,AANDB,SP15 ; HIGH COUN
2920 010252 MEM MEMX,ADDC,SP15,INCMAR ; OVERALL
2921 ; COUNT
2922 010254 BRWRTI IMM,16 ; BIT COUNT
2923 010256 SP BR,AANDH,SP3,INCMAR ; AND
2924 010260 BRWRTI IMM,EOF ; END OF
2925 010262 SF BR,AORB,SP3 ; FRAME
2926 010264 ALWAYS HIPAAD
2927 ;
2928 ;
2929 010266 DMAC10: BRWRTI IMM,27 ; NOT LAST BUFFER
2930 010270 SP BR,AANDB,SP15 ; HIGH COU
2931 010272 MEM MEMX,ADDC,SP15,INCMAR ; OVERALL
2932 010274 BRWRTI IMM,0,INCMAR ; COUNT
2933 010276 SP BR,SELB,SP3 ; ALL CASES HIGH PART
2934 ; OF ADDRESS
2935 ;
2936 ; CONTINUE NEXT PAGE
  
```

```
2938 ;
2939 ;
2940 ; *****
2941 ; * HANDLING OF HIGH PART OF ADDRESS *
2942 ; *****
2943 ;
2944 ;
2945 010300 HIPAAD: SP IRUS,INDAT2,SP8
2946 010302 HIPA10: BRWETE I*M,300
2947 010304 BRWETE BR,AANDB,SP8
2948 010306 BRUTAT
2949 010310 BRUTAT
2950 010312 BRUTAT
2951 010314 BRUTAT
2952 010316 SP BR,SELB,SP8
2953 010320 RTNSUB SPS,P1
2954 ;
2955 ;
```

2957  
 2958  
 2959  
 2960  
 2961  
 2962  
 2963  
 2964  
 2965  
 2966  
 2967  
 2968  
 2969  
 2970  
 2971  
 2972  
 2973  
 2974  
 2975 010322  
 2976 010324  
 2977 010326  
 2978 010330  
 2979 010332  
 2980 010334  
 2981 010336  
 2982 010340  
 2983 010342  
 2984  
 2985 010344  
 2986 010346  
 2987  
 2988 010350  
 2989 010352  
 2990 010354

```

;
; *****
; * LOAD CONTEXT *
; *****
;
; THIS ROUTINE LOADS CONTEXT FOR
; TRANSFERRING DATA, AFTER AN
; INTERRUPTION
; THE CONTEXT IS : SAVED DATA      (2 BYT)
;                   COUNT           (2 BYT)
;                   BUFFER ADDRESS  (3 BYT)
; INIT BLOCK COUNT
;
; THE MAR POINTS ON BLOCK ADD + 7 AFTER
;
; THE RETURN ADDRESS IS IN SP5
;
;
;
LOCONT: BPWRTI IMM,10                ; INIT
        MODST  BR,ADD,SP1,LDMAR      ; MAR
        OUT    MEMX,SELB,OUTDA2,INCMAR ; DATA
        OUT    MEMX,SELB,OIDAT2,INCMAR
        SP     MEMX,SELB,SP14,INCMAR  ; COUNT
        SP     MEMX,SELB,SP15,INCMAR
        OUT    MEMX,SELB,IBA1,INCMAR  ; ADDRESS
        OUT    MEMX,SELB,IBA2,INCMAR
        OUT    MEMX,SELB,OUTDA1
;
        BPWRTI IMM,BLOCK              ; INIT BLOCK COUNT
        SP     RE,SELB,SP9
;
        BPWRTI IMM,7                  ; RELOADMAR
        MODST  BR,ADD,SP1,LDMAR
        RTNSUB SP5,P1
    
```

2992  
 2993  
 2994  
 2995  
 2996  
 2997  
 2998  
 2999  
 3000  
 3001  
 3002  
 3003  
 3004  
 3005  
 3006  
 3007 010356  
 3008 010360  
 3009 010362  
 3010 010364  
 3011 010366  
 3012 010370  
 3013  
 3014 010372  
 3015 010372  
 3016 010374  
 3017 010376  
 3018 010400  
 3019 010402  
 3020  
 3021 010404  
 3022 010406  
 3023  
 3024 010410  
 3037

```

;
;
; *****
; END OF FRAME DETECTED BY LJNE UNIT RX *
; *****
;
;
; IF RX STATE IS : INFO FRAME, NO BUFFER,
; THEN LOAD ERROR = NO BUFFER AVAILABLE
; NEW STATE = RX ABORT
; GO TO ROUTINE THAT EMPTIES THE FIFO
; ELSE, EXIT
;
;
;
FS00: LDMAP IMM,1 ; RECEPTION
      SP MEMX,SELB,SP10,INCMAR ; TEST RX STATE
      BRWPT IMM,INOB,INCMAR
      MODST HF,SUBOC,SP10,INCMAR
      Z FS10
      ALWAYS SC30 ; INFO, NO BUFFER
; ; EXIT
FS10: BRWPT MEMX,SELB
      BRO SC30 ; TEST IF BUFFER
      BRWPT IMM,17 ; BUFFER : EXIT
      MODST LF,ADD,SP1,LDMAR ; ERROR
      MEM IMM,NOBUF
; ; NO BUFFER AVAILABLE
      BRWPT IMM,RXINFO
      SP HF,SELB,SP2
; ; TYPE OUT IN SP2
;
ALWAYS FI1255
    
```

\*KMC\* BREAKPOINTS DEFINITION

```

1 000000
2 010412
3 010514
4 000000
5 000004
6 010412
7 010514
8 000000
9 000004
10
11
12
13
14
15
16
17
18
19 010534 010534
20
21 000001

```

KMCODE: KMCODE

.SBTTL \*KMC\* BREAKPOINTS DEFINITION

```

.NPKKS = 4
.SAV=.
.=KSTAKT+<<1024.-<NPKKS*2>>*2>
SNBRKS=0
.PEPT NPKKS
.IRF SSNDHh,<\SNBRKS>
OUT IMM,SSNBK*20+1,OMAIN
ALWAYS .
SNBRKS=SNBRKS+1
.ENDM
.ENDM
.END

```

```

000000
000004
010412
010514
000000
000004
010412
010514
000000
000004
010534
010534
000001

```



AANIB = 000260	BAR2 = 003740R	DFBUT1 = 004536R	F11500 = 007066R	INTRDO = 003020K
ARUP = 000012	BEBUOP = 000003	DECA = 060160	F11510 = 007070R	INVCOM = 000040
ARCFEF = 000013	HFGFRA = 000002	DISCO = 000003	F11520 = 007072R	INIE = 000020
AROKR = 000014	HFGIN = 000000R	DISP = 000234	F11530 = 007074R	INIEEF = 000021
ABURRF = 000015	BFEF = 000001	DMAC = 010224R	F11540 = 007106R	IN10 = 000022
ABUR1 = 000100	BIT0 = 000001	DMACR = 006144R	F11600 = 067124R	IN10EF = 000023
ARUF1R = 003779R	BIT1 = 000002	DMAC10 = 010266R	F11610 = 007132R	IN2E = 000024
ABUF00 = 003776R	BIT10 = 002000	DM100 = 000306R	F11700 = 007146R	IN2EEF = 000025
ACSR = 000052R	BIT11 = 004000	DM101 = 000624R	F11900 = 007152R	IOBA1 = 000140
ACSREX = 004004R	BIT12 = 010000	DM102 = 001222R	F12000 = 007162R	IOBA2 = 000160
ADD = 000000	BIT13 = 020000	DV101 = 000014R	F12100 = 007170R	IODAT1 = 000040
ADDC = 000020	BIT14 = 040000	ECSP = 000000	F12200 = 007176R	IODAT2 = 000060
ADDCON = 004012R	BIT15 = 100000	ENDPSS = 104402	F12210 = 007216R	ITIN = 000200
ADDR = 000006R	BIT2 = 000004	ENDS = 104403	F12220 = 007232R	ITOUT = 000300
ALCOND = 000400	BIT3 = 000010	EOF = 000240	F12300 = 007240R	JUMP = 100000
AMASK = 000022	BIT4 = 000020	ERGEN = 007554R	F12400 = 007246R	KFREE = 000000
AMES00 = 003274R	BIT5 = 000040	ERGF00 = 007700R	F12410 = 007262R	KMCODE = 010534R
AMES01 = 003302R	BIT6 = 000100	ERKMC = 007466R	FREE = 000104R	KSTART = 004534R
AMES02 = 003310R	BIT7 = 000200	ERKMC A = 007476R	FS00 = 010356R	LDMAPG = 004000
AMES03 = 003316R	BIT8 = 000400	ERKMC E = 007500R	FS10 = 010372R	LDMAR = 010000
AMES04 = 003322R	BIT9 = 001000	ERKMC1 = 007662R	GETPAS = 104413	LFADDR = 000014
AMES05 = 003330R	BLOCK = 000040	ERKMC2 = 007664R	GOOD = 003732R	LFCIRL = 000013
AMES06 = 003336R	BOP = 000300	ERMAS = 007736R	GR0C = 000015	LFDAT1 = 000012
AMES07 = 003344R	BR = 060000	ERMAS1 = 007736R	GR0U = 003174R	LFDAT2 = 000011
AMES08 = 003350R	BREAKS = 104407	ERRCNT = 000030R	G*BUFS = 104412	LFDAT3 = 000010
AMES09 = 003356R	BR0CON = 002000	ERRNS = 104410	HIPAAD = 010300R	LFD1 = 000004
AMES10 = 003364R	BR1 = 000012R	ERRORS = 104404	HIPAI0 = 010302R	LFD1EF = 000005
AMES11 = 003372R	BR1CON = 002400	EXITS = 104400	IBA1 = 002004	LFD2 = 000006
AMES12 = 003400R	BR2 = 000013R	FCSEKK = 000040	IBA2 = 002005	LFD2EF = 000007
AMES13 = 003404R	BR4CON = 003000	FICHAR = 000005	IBUS = 020000	LFD3 = 000010
AMES14 = 003412R	BR7CON = 003400	FIFCTR = 000020	IB1 = 000026	LFD3EF = 000011
AMES15 = 003420R	BUFIN = 004134R	FI0000 = 006524R	IB1EF = 000027	LIN = 003742R
AMES16 = 003424R	BUFKMC = 004114R	FI0100 = 006560R	IB2 = 000030	LINAUT = 002100R
ADRB = 000300	BUFDVE = 000002	FI0200 = 006564R	IB2EF = 000031	LINENM = 100060
ADRN = 000240	BYCO = 004020R	FI0210 = 006602R	IDLE = 000000	LINUMB = 003744R
APLUSC = 060100	CALLPC = 004026R	FI0300 = 006606R	IE1 = 000032	LOADER = 002312R
ASB = 000056R	CCOND = 001000	FI0400 = 006616R	IE1EF = 000033	LOCONT = 010322R
ASEL0 = 004050R	CDATAS = 104414	FI0500 = 006630R	IE2 = 000034	LOFICO = 002014
ASEL10 = 004060R	CLEKMC = 002420R	FI0600 = 006636R	IE2EF = 000035	LOFIDA = 002015
ASEL12 = 004062R	CLK = 004014R	FI0800 = 006650R	IIBA1 = 000100	LOLUAD = 002010
ASEL14 = 004064R	CLK1 = 004016R	FI0900 = 006662R	IIBA2 = 000120	LOPCN = 004024R
ASEL16 = 004066R	COMM = 000007	FI1000 = 006666R	IMM = 000000	LOSTHI = 002017
ASEL2 = 004052R	COMMAN = 000007	FI1100 = 006672R	IMODX = 000000	LOSTLO = 002016
ASEL4 = 004054R	CSRA = 000050R	FI1110 = 006674R	INCA = 060060	LUACO = 007446R
ASEL6 = 004056R	CSREX = 004104R	FI1120 = 006700R	INCMAR = 014000	LUBUSY = 000200
ASTA1 = 000054R	CTRLER = 000003	FI1200 = 006714R	INCON = 100000	LUERO = 007372R
AWAS = 000060R	DATA12 = 003746R	FI1220 = 006732R	INDAT1 = 000000	LUER10 = 007410R
AXURB = 000320	DATA3 = 003750R	FI1221 = 006740R	INDAT2 = 000020	LUER30 = 007420R
BAD = 003734R	DAT1CKS = 104417	FI1225 = 006754R	INFO1 = 003752R	LUER40 = 007426R
BADA = 000265	DATERs = 104405	FI1230 = 006770R	INFO2 = 003762R	LUF10 = 005206R
BAD1N = 010134R	DATI = 000001	FI1240 = 006772R	INIT = 000022R	LUF120 = 005250R
BAD1N0 = 010160R	DATI H = 000003	FI1245 = 007002R	INOH = 000016	LUF130 = 005254R
BAD1N1 = 010210R	DATI0 = 000021	FI1250 = 007022R	INOPFE = 000017	LUF140 = 005260R
BAD1N2 = 010216R	DATI0B = 000221	FI1255 = 007030R	INSTHI = 003731R	LUF150 = 005344R
BAD1N3 = 010152R	DATI0BH = 000223	FI1260 = 007040R	INSTLO = 003730R	LUF151 = 005346R
BANOLD = 000020	DATI0H = 000023	FI1300 = 007050R	INIFLG = 003724R	LUF152 = 005356R
BAR1 = 003736R	DFBUT = 004534R	FI1400 = 007060R	INTRDI = 002674R	LUF153 = 005374R

LUF157 005404K	MODNAM= 000000K	PEA20 005712K	PSW = 177776	SC417 005136R
LUF160 005410K	MODSP = 000206K	PEA30 005716K	PUSH = 005746	SC418 005142R
LUF161 005412K	MODSTA= 000003	PEC00 005740K	PUSH2 = 024646	SC42 005150R
LUF162 005416K	MOST = 000003	PFC10 005742R	P0 = 000000	SC43 005160R
LUF163 005416K	MOVF = 000000	PPC20 005750K	P00 = 004716K	SC430 005174R
LUF164 005424K	MSGMS = 104411	PPD00 005772K	P1 = 004000	SC44 005200R
LUF165 005454K	MSGSS = 104416	PRD10 006002K	P10 = 004754R	SC50 005016R
LUF168 005466K	MSG = 104406	PRD20 006010K	P2 = 010000	SECHAR= 000004
LUF169 005474K	MSTIN = 000357	PRL30 006016K	P20 = 005560R	SELA = 060200
LUF00 007276K	MSTOU = 000367	PPD40 006032K	P3 = 014000	SELB = 000220
LUF010 007314K	NBRKS = 000004	PRD50 006042K	UES = 104401	SFADDR= 000006
LUF020 007320K	NOBAD 007506K	PRD60 006062K	RANDOM 003040K	SFCTRL= 000007
LUF030 007356K	NOBUFF= 000001	PRD70 006072K	RANNAD 003034K	SHFTBR= 001400
LUF040 007364K	NOEXME= 000004	PRD80 006102K	RANNUM 003036K	SMAS = 000001
LUF10 007454K	NORM = 000010	PRD90 006110K	RBUFEA 000070R	SPBRX = 003400
LUMSU 007440K	NOIPRE= 000010	PRL95 006116K	RBUFPA 000066K	SPOINT 000024R
LU0 004550K	NPR = 100200	PRD98 006126K	RBUFSZ 000072K	SPSIZ = 000040
LU10 004576K	ORA1 = 002006	PR00 005564R	RBUFVA 000064R	SPX = 003000
LU20 004604K	OBA2 = 002007	PR0010 005612R	RDY1 = 000200	SPO = 000000
LU30 004614K	ORR = 001011	PCA00 006170R	RDY0 = 000020	SP1 = 000001
LU40 004636K	OCON = 100040	PCB00 006206K	REFICO= 000300	SP10 = 000012
MAIN = 100020	OIDA11= 002000	PCC00 006222K	REFIDA= 000320	SP11 = 000013
MA11 = 000011	UIDAT2= 002001	PCC00 006230K	RELUAD= 000200	SP12 = 000014
MAP22S= 104415	OINCOM= 001000	PC00 006146K	RESCAN= 000240	SP13 = 000015
MASK 003726K	OKMES 007516K	PDA00 006266K	RESTHI= 000360	SP14 = 000016
MC11 = 000347	OKMESQ 007540R	PDB00 006300K	RESTLO= 000340	SP15 = 000017
MC12 = 000020	OKMESR 007530R	PD00 006250K	RESIRT 000250K	SP16 = 000016
MC13 = 000377	OKMESS 007522K	PE00 006370R	RQCSR 002600R	SP17 = 000017
MC21 = 000337	OKMOD 007546K	PE10 006404K	RQCSR1 002626K	SP2 = 000002
MC23 = 000375	OKMS 007650K	PE20 006412K	RQCSK3 002672R	SP3 = 000003
MC31 = 000376	OLINEM= 001003	PEA00 006436K	RSTKI 000062R	SP4 = 000004
MC33 = 000377	OMAIN = 001001	PFB00 006442K	RXD1 = 000007	SP5 = 000005
MC41 = 000075	ONPR = 001010	FFB10 006444K	RXDMA0 010046R	SP6 = 000006
MC43 = 000306	OCON = 001002	PF00 006424R	RXDMA1 010056R	SP7 = 000007
MEM1 = 054000	OPEN = 000000	PG00 006460K	RXDMA2 010070R	SP8 = 000010
MEMX = 040000	OPORT1= 001004	PG10 006500K	RXDMA3 010100R	SP9 = 000011
MESCU 007574R	OPORT2= 001005	PH00 006512K	RXFREE= 000000	SR1 000016R
MESCU1 007566K	OPORT3= 001006	PHU1 006520K	RXIN = 000000	START 000206R
MESCSR 007736K	OPORT4= 001007	PJRS = 000004	RXINFO= 000000	STAT 000020R
MESE 003064K	OUTDA1= 002002	PJ00 005524K	RXLLOC= 000002	STSCAN= 002012
MESS00 003430K	OUTDA2= 002003	EK00 005504K	RXON = 000004	SUB = 000340
MESS01 003437K	OUTME1 007620K	EK01 005512K	RXSHCO= 000001	SUBC = 000040
MESS02 003446K	OXREG0= 002010	POPSP = 005726	R6 = 000006	SUBOC = 000360
MESS03 003455K	OXREG1= 002011	POPSP2= 022626	F7 = 000007	SVR0 000032R
MESS04 003501R	OXREG2= 002012	PORT1 = 100100	SHADR 000052R	SVR1 000034R
MESS05 003515K	OXREG3= 002013	PORT2 = 100120	SCREL 007274R	SVR2 000036R
MESS06 003531K	OXREG4= 002014	PORT3 = 100140	SC00 004664R	SVR3 000040R
MESS07 003550K	OXREG5= 002015	PORT4 = 100160	SC28 004656R	SVR4 000042R
MESS08 003566K	OXREG6= 002016	PPTY0 = 000000	SC30 004702R	SVR5 000044R
MESS09 003603K	OXREG7= 002017	PPTY1 = 000040	SC31 004706R	SVR6 000046R
MESS10 003630K	PASCNT 000026R	PPTY2 = 000100	SC40 005022R	TABAD 007774R
MESS11 003640K	PASS 002074R	PPTY3 = 000140	SC409 005060R	TABAD1 010020R
MESS12 003650K	PA00 006322K	PPTY4 = 000200	SC410 005042R	TABAD2 010032R
MESS13 003664K	PA001 006354K	PPTY5 = 000240	SC411 005066K	TABAR1= 000042R
MESS14 003703K	PA002 006386K	PPTY6 = 000280	SC412 005090K	TABAR2= 000044R
MF11 = 000277	PA003 006418K	PR117 000340	SC413 005114K	TABAR3= 000046R
MELD = 000327	PA015 006700R	PR 07776	SC414 005138K	TABAR4= 000048R
				TABAR5= 000050R
				TABAR6= 000052R
				TABAR7= 000054R
				TABAR8= 000056R
				TABAR9= 000058R
				TABAR10= 000060R
				TABAR11= 000062R
				TABAR12= 000064R
				TABAR13= 000066R
				TABAR14= 000068R
				TABAR15= 000070R
				TABAR16= 000072R
				TABAR17= 000074R
				TABAR18= 000076R
				TABAR19= 000078R
				TABAR20= 000080R
				TABAR21= 000082R
				TABAR22= 000084R
				TABAR23= 000086R
				TABAR24= 000088R
				TABAR25= 000090R
				TABAR26= 000092R
				TABAR27= 000094R
				TABAR28= 000096R
				TABAR29= 000098R
				TABAR30= 000100R

SYMBOL TABLE

```

XPI = 000000
XPA = 000020
XQA = 000120
XQAC = 000140
XAPOR = 000015
XDI = 000006
XLI = 000004
XINFO = 000004
XIN1 = 010110F
XIN11 = 010124F
XIN12 = 010130F
XIN13 = 010120F
XLD = 000006
XLDC = 000006
XNDE = 000016
XOH = 000005
XSH = 000005
XSHC = 000005

```

```

XK00 = 00032F
1YPCUM = 004022H
100002 = 000326H
100003 = 000350H
100004 = 000354F
100005 = 000360H
100006 = 000420R
100007 = 000466R
100010 = 000516H
100011 = 000546R
100012 = 000572H
100013 = 000610R
101002 = 000644R
101003 = 000666R
101004 = 000672R
101005 = 000676R

```

```

101006 = 00071F
101007 = 000744F
101010 = 001074F
101011 = 001124H
101012 = 001154H
101013 = 001170R
101014 = 001206H
102002 = 001230R
102003 = 001276R
102004 = 001320R
102005 = 001324R
102011 = 001604R
102014 = 001652R
102015 = 001702R
102016 = 001732H
102017 = 001752H
102020 = 001770H

```

```

102022 = 001770H
102024 = 001804R
102030 = 002062R
UFBF = 100220
VFCTUR = 000010R
WASADR = 000054R
WBUFEA = 000076R
WBUFPA = 000074R
WBUFRQ = 000100R
WBUFSZ = 000102R
WRMEM = 002400
WROUT = 002000
WROU1X = 001000
WRTFBF = 000400
WTANS = 002710R
W1ANS1 = 002726R
W1ANS2 = 002744R

```

```

W1A1 = 000020
W1A2 = 000020
W1A3 = 000020
W1A4 = 000020
W1A5 = 000020
W1A6 = 000020
W1A7 = 000020
W1A8 = 000020
W1A9 = 000020
W1A10 = 000020
W1A11 = 000020
W1A12 = 000020
W1A13 = 000020
W1A14 = 000020
W1A15 = 000020
W1A16 = 000020
W1A17 = 000020
W1A18 = 000020
W1A19 = 000020
W1A20 = 000020

```

```

ABS. 000000 000
010536 001

```

ERRORS DETECTED: 0

```

VIRTUAL MEMORY USED: 9584 WORDS ( 38 PAGES)
DYNAMIC MEMORY: 10254 WORDS ( 39 PAGES)
ELAPSED TIME: 00:04:07

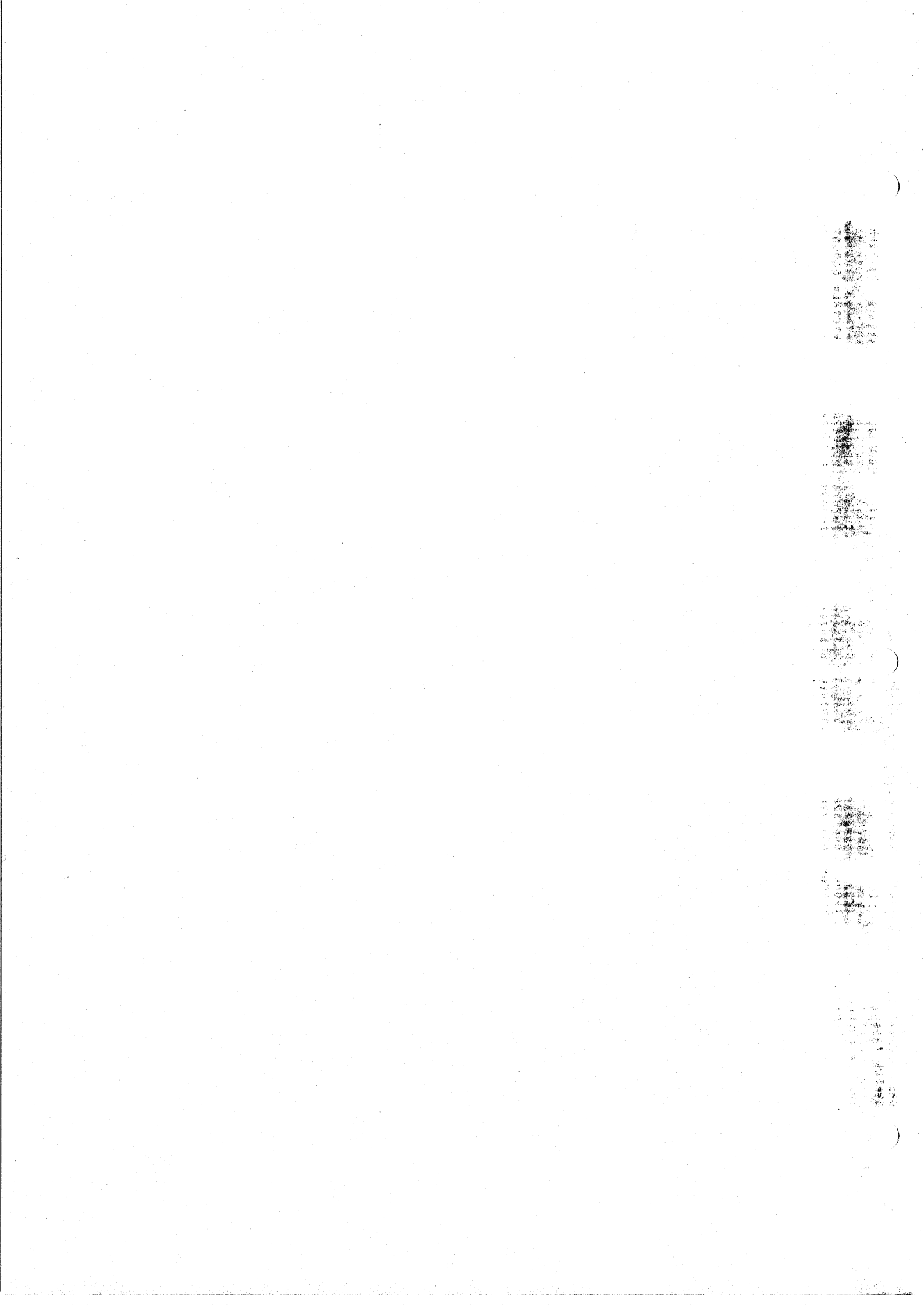
```

[Faded dump text containing symbols and values]

```

...
XPA = 000020
XQA = 000120
XQAC = 000140
XAPOR = 000015
XDI = 000006
XLI = 000004
XINFO = 000004
XIN1 = 010110F
XIN11 = 010124F
XIN12 = 010130F
XIN13 = 010120F
XLD = 000006
XLDC = 000006
XNDE = 000016
XOH = 000005
XSH = 000005
XSHC = 000005
...

```





**digital**

**Computer Special Systems**