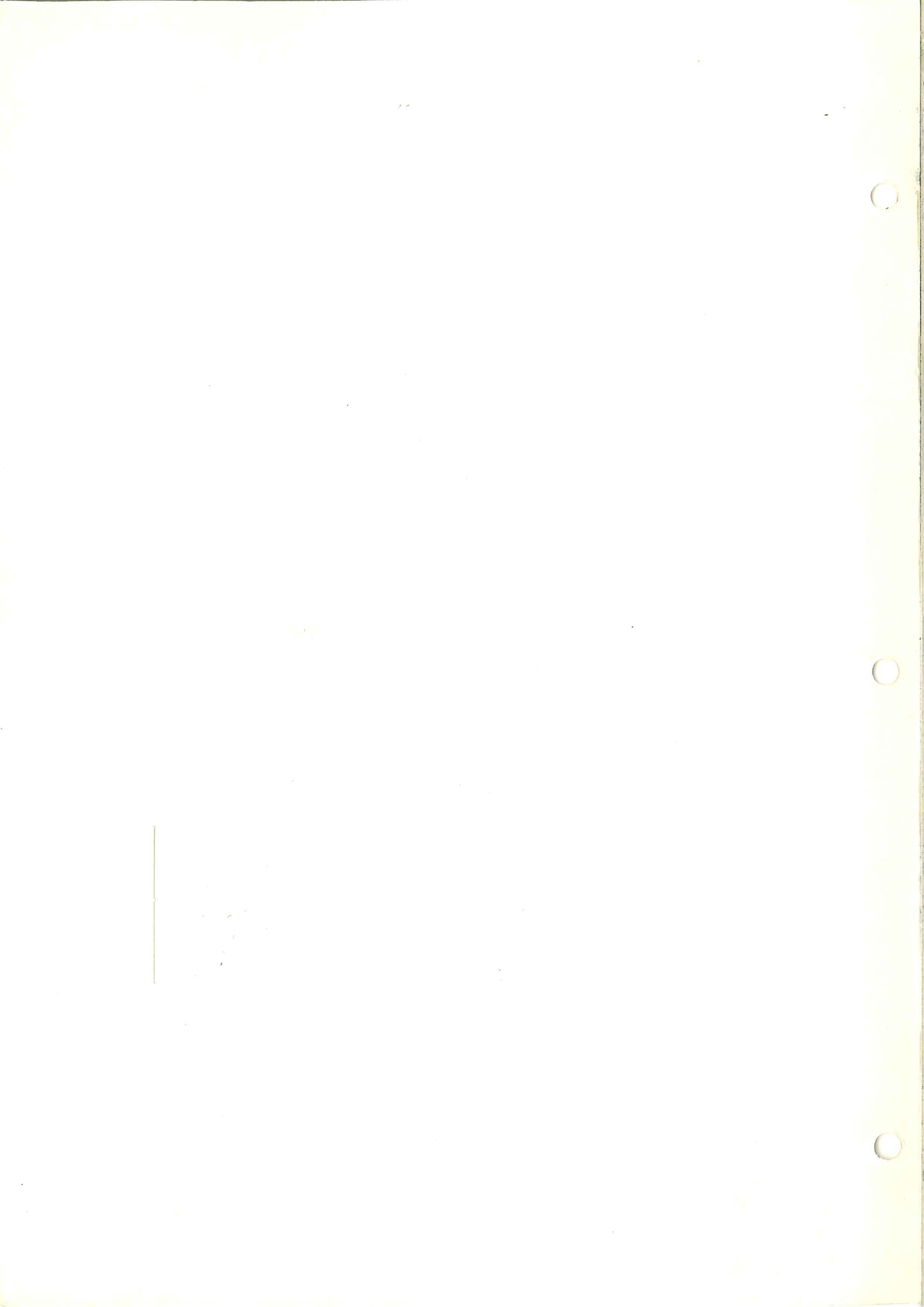


digital
digital
digital
digital
digital
digital
digital

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

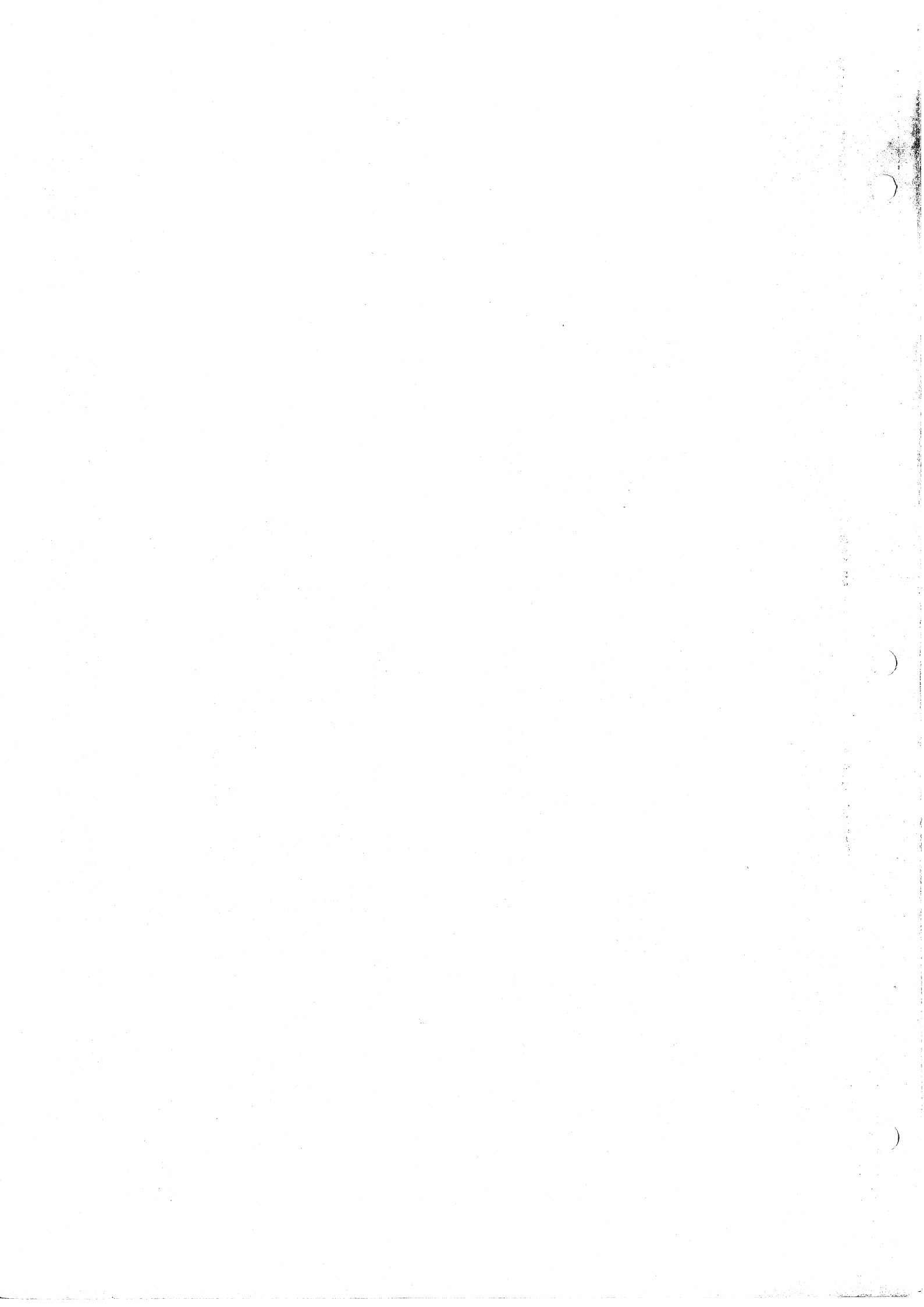
ANNECY

SECRET

CONFIDENTIAL

CONFIDENTIAL

SECRET



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 INTERPUPT)	
17-	640	WAIT KMC ANSWER (RDYU INTERRUPT)	
18-	685	RANDOM NUMBER GENERATOR SUBROUTINE	
19-	703	ERFUR MESSAGE FROM KMC	
20-	734	SPECIAL COMMANDS	
21-	770	ASCII STRINGS	
23-	816	PROGRAM VARIABLES	
24-	857	BUFFERS	
37-	2	**KMC*	**** DMS11F KMC MJCRCODE ****
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-	500	**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 RECEPITION
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

•REV-

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).

74
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
DEVIC 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
000000

LOMODX <DMSA >,160300,330,5,0,0,BUFIN,200,200
MODULE 150000,DMSA,160300,330,5,0,0,BUFIN,200,200
.TITLE DMSA DEC/X11 SYSTEM EXERCISER MODULE
DDXCOM VERSION 4(CSSUK) 14-FER-77
;*****
;*****

BEGIN:
MODNAM: .ASCII /DMSA / ;MODULE NAME.
115
040

123 XFLAG: .BYTE UPEN ;TU KEEP TRACK OF WBUFF USAGE
ADDR: 160300+0 ;1ST DEVICE ADDRESS
VECTOR: J30+U ;1ST DEVICE VECTOR
BR1: .BYTE PRTY5+0 ;1ST DEVICE BR LEVEL
BR2: .BYTE PRTY0+0 ;2ND DEVICE BK LEVEL
DVID1: 0+1 ;DEVICE INDICATOR 1
SR1: OPEN ;SWITCH REGISTER 1
;*****
;*****

STAT: 150000 ;STATUS WORD
INIT: START ;MODULE START ADDRESS
SPOINT: MODSP ;MODULE STACK POINTER
PASCNT: 0 ;PASS COUNTER
ERRCNT: 0 ;ERROR COUNTER
SVR0: OPEN ;LOC TO SAVE R0
SVR1: OPEN ;LOC TO SAVE R1
SVR2: OPEN ;LOC TO SAVE R2
SVK3: OPEN ;LOC TO SAVE R3
SVK4: OPEN ;LOC TO SAVE R4
SVK5: OPEN ;LOC TO SAVE R5
SVK6: OPEN ;LOC TO SAVE R6
CSKA: OPEN ;ADDRESS OF CURRENT CSR
SBADR: OPEN ;ADDRESS OF GOOD DATA, OR
ACSR: OPEN ;CONTENTS OF CSR.
WASADR: OPEN ;ADDRESS OF BAD DATA, OR
ASAT: OPEN ;STATUS REGISTER CONTENTS.
ASB: OPEN ;EXPECTED DATA.
AWAS: OPEN ;ACTUAL DATA.
RSIRT: RESTRT ;RESTART ADDR AFTER END OF PASS
RBUFVA: BUFIN ;READ BUFFER VIRTUAL ADDR.
RBUFPA: OPEN ;READ BUFFER PHYSICAL ADDR.
RBUFEA: OPEN ;READ BUFFER EA BITS.
RBUFSZ: 200 ;SIZE OF THE READ BUFFER
WBUFPA: OPEN ;WRITE BUFFER PHYSICAL ADDR.
WBUFEA: OPEN ;WRITE BUFFER EA BITS
WRUFRO: 200 ;WRITE BUFFER SIZE REQUESTED
WBUFSZ: OPEN ;WRITE BUFFER SIZE AVAILABLE
FREE: OPEN ;RESERVED FOR FUTURE USE
;*****
;*****

000020 150000
000022 000206
000024 000206
000026 000000
000030 000000
000032 000000
000034 000000
000036 000000
000040 000000
000042 000000
000044 000000
000046 000000
000050 000000
000052 000000
000054 000000
000056 000000
000060 000000
000062 000250
000064 004134
000066 000000
000070 000000
000072 000200
000074 000000
000076 000000
000100 000200
000102 000000
000104 000000

.REPT SPSI2
.NLIST 0
.WORD
.LIST
.FNDR

MODSP:
;*****
;*****

000206

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

AUTOCONFIGURATION

.SBTIL AUTOCONFIGURATION

```

129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156

; BASE DEVICE ADDRESS =
; SEL0 ADDRESS
; DEVICE ADDRESS + 2 =
; SEL2 ADDRESS
; DEVICE ADDRESS + 4 =
; SEL4 ADDRESS
; DEVICE ADDRESS + 6 =
; SEL6 ADDRESS
; AUTO-CONFIGURATION
; LOAD KMC MICROPROGRAM

; BASE VECTOR ADDRESS
; 1ST INTERRUPT ROUTINE
; SET BR LEVEL

; 2ND INTERRUPT ROUTINE
; SET BR LEVEL

; SAVE THE CSR ADDRESS

*****
* INITIALISATION *
*****

ADDR,R0
R0,ASELO
(F0)+
R0,ASEL2
(R0)+
R0,ASEL4
(R0)+
R0,ASEL6
PC,LINAUT
PC,LOADER

VECTOR,R0
#INTEDI,(R0)+
BR1,(R0)+
(R0)+
#INTRDO,(R0)+
BR1,(R0)+
(F0)+
ASELO,CSRA

START:
MOV 177574
MOV 003632
TST 005720
MOV 003626
TST 010067
MOV 003622
TST 010067
MOV 003616
JSR 001634
JSR 002042

; RESTART:
MOV 177534
MOV 002674
MOV 177526
CLR 105020
MOV 003020
CLR 177514
MOV 003544
MOV 177542

```

Line	Address	Instruction	Label	Description
158				.SBYTL TEST 0
159				DESCRIPTION:
160				TRANSMISSION AND RECEPTION IN A
161				LOOPING BACK OF A SHORT CONTROL
162				FRAME
163				INITIALIZATION OF THE LJNE UNIT
164				TRANSMISSION OF A RANDOM FRAME
165				WAITING FOR TWO MESSAGES FROM KMC
166				- ACKNOWLEDGE OF THE TRANSMISSION
167				- END OF RECEPTION
168				CHECKING OF THE CONTENT OF THE TWO FRAMES
169				
170				
171				
172				
173	000306	MOV	DMT00:	BAR1, BAR2 ; LINE MASK
174	000314	MOV		#1, L1N ; SELECT LINE 0
175	000322	CLR		L1NUMB ; LINE NUMBER = 0
176				
177	000326	MOV	T00002:	#400, ADDCON ; RETURN VALUE
178	000334	CLR		LOOPCN ; CLEAR LOOP COUNT
179	000340	BIT		L1N, BAR2 ; IS THE LINE PRESENT ?
180	000346	BEQ		T00013 ; NO : NEW LINE
181				
182	000350	JSR	T00003:	PC, CLEKMC ; INIT KMC
183				
184	000354	JSR	T00004:	PC, ROGSK ; REQUEST CSR REGISTER
185				
186	000360			
187	000364	JSR	T00005:	PC, RANDOM ; AVOID 0 IN AD CHARACTER
188	000372	MOVE		RANUM, ADDCON ; YES : ANOTHER ONE
189		BEQ		T00005
190	000374	MOV		ADDCON, @ASEL4 ; LOAD MESSAGE
191	000402	MOV		ADDR, R1 ; KMC BASE ADDRESS
192	000406	MOVB		L1NUMB, 3(R1) ; LINE NUMBER
193	000414	MOVB		#TXSH, (R1) ; SHORT CONTROL
194				
195	000420	JSR	T00006:	PC, WTANS ; WAIT ANSWER FROM KMC
196	000424	CMPB		#25, @ASEL2 ; END OF TX ?
197	000432	BEQ		T00007 ; YES
198	000434	CMPB		#21, @ASEL2 ; END OF RX ?
199	000442	BEQ		T00010
200				
201	000444	JSR		PC, MESE ; MODEM STATUS
202	000450			ERRNS, BEGIN, TABMOD ; *****
203	000456			MSGSS, AMES00, BEGIN ; *****
204	000464	BR		T00003 ; ASCII MESSAGE CALL WITH NO HEADER
205				
206	000466	TST	T00007:	@ASEL2 ; ERROR FLAG SET ?
207	000472	BGE		T00006 ; NO : TRY AGAIN
208				
209	000474	JSR		PC, MESE ; MODEM STATUS
210	000500			ERRNS, BEG-3, TABMOD ; *****

TEST 0

```

211 000506 104416 003302 000000
212 000514 000715
213
214 000516 005777 003330
215 000522 002011
216
217 000524 004767 002334
218
000530 104410 000000 004030
219 000536 104416 003310 000000
220 000544 000412
221
222 000546 026777 003240 003300
223 000554 001406
224
000556 104410 000000 004050
225 000564 104416 003316 000000
226
227 000572
000576 104407 000000
000602 104407 000000
000606 105367 003216
000606 001262
230
231 000610 005267 003130
232 000614 000241
233 000616 006367 003120
234 000622 103241

;*****
MSGSS,AMES01,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
BP T00003 ; TPY AGAIN
;
T00010: TST @ASFL2 ; ERROR FLAG SET ?
BGE T00011 ; NO : O.K.
;
JSK PC,MESE ; MODEM STATUS
;*****
ERRNS,BEGIN,TABMOD ;
;*****
MSGSS,AMES02,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
BK T00012 ; END
;
T00011: CMP ADDCON,@ASEL4 ; COMPARE RECEIVED DATA
BEQ T00012 ; O.K.
;*****
ERRNS,BEGIN,TABKMC ;
;*****
MSGSS,AMES03,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
;
T00012:
BREAKS,ELGIN ; TEMPORARY RETURN TO MONITOR...
BREAKS,BEGIN ; CONTINUE AT NEXT INSTRUCTION.
DECB LOOPCN ; LOOP COUNT
BNE T00004 ; # 0 : LOOP
;
T00013: INC LINUAB ; NEW LINE NUMBER
CLC ;
ASL LIN ; SHIFT TEST MASK
BCC T00002 ; NOT LAST L.U. , LOOP

```

TEST 1

Line No.	Address	Instruction	Operand 1	Operand 2	Operand 3	Description
236						.SHTIL TEST 1
237						DESCRIPTION:
238						TRANSMISSION AND RECEPTION IN A
239						LOOPING BACK OF A LONG CONTROL
240						FRAME
241						INITIALIZATION OF THE LINE UNIT
242						TRANSMISSION OF A RANDOM FRAME
243						WAITING FOR TWO MESSAGES FROM KMC
244						- ACKNOWLEDGE OF THE TRANSMISSION
245						- END OF RECEPTION
246						CHECKING OF THE CONTENT OF THE TWO FRAMES
247						
248						
249						
250						
251	000024	MOV	003106	003106		BAR1, BAR2 ; LINE MASK
252	000032	MOV	003102	003102		#1, LIN ; SELECT LINE 0
253	000040	CLK				LINUMB ; LINE NUMBER = 0
254						
255	000644	MOV	003140	003140		#103400, ADDCON ; ANSWER
256	000652	CLR				LOOPCN ; CLEAR LOOP COUNT
257	000656	BIT	003054			LIN, BAR2 ; IS THE LINE PRESENT
258	000664	REQ				T01014 ; NO : NEW LINE
259						
260	000666	JSR		001526		PC, CUEKMC ; INIT KMC
261						
262	000672	JSK		001702		PC, RUCSR ; REQUEST CSR REGISTER
263						
264	000676					
265	000702	JSK		003102		PC, RANDOM ; AVOID 0 IN AD CHARACTER
266	000710	REQ				T01005 ; YES : ANOTHER ONE
267						
268	000712	MOV	003074	003134		ADDCON, @ASEL4 ; LOAD IN SEL4
269	000720	JSR				PC, RANDOM
270	000724	MOV	003126	003126		RANUM, @ASEL10 ; LOAD IN SEL10
271	000732	MOV	003006	003006		RANUM, DATA12 ; SAVE VALUE
272	000740	JSR				PC, RANDOM
273	000744	MOV	003110	003110		RANUM, @ASEL12 ; LOAD IN SEL12
274	000752	MOV	002770	002770		RANUM, DATA3 ; SAVE VALUE
275	000760	MOV				ADDR, R1 ; KMC BASE ADDRESS
276	000764	MOV	000003	000003		LINUMB, 3(R1) ; LINE NUMBER
277	000772	MOV				#TXLD, (R1) ; LONG CONTROL
278						
279	000776	JSR				PC, WTANS ; WAIT ANSWER FROM KMC
280	001002	CMPB		003042		#26, @ASEL2 ; END OF TX ?
281	001010	BEG				T01007 ; YES
282	001012	CMPH		003032		#22, @ASEL2 ; END OF RX ?
283	001020	BEG				T01010 ; YES
284						
285	001022	JSR		002036		PC, MESE ; MODEM STATUS
286						*****
287	001034	ERRMS		004030		BEGIN, TABMOD ;
288	001042	MSGSS		003322		AVES04, BEGIN ; ASCII MESSAGE CALL WITH NO HEADER
289		BR				T01003 ; TRY AGAIN

TEST 1

```

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,PESE ; MODEM STATUS
294 001056 104410 000000 004030 ;*****
;ERRNS,BEGIN,TABMOD ;*****
;*****
;MSGSS,AMES05,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
BR T01003 ; TRY AGAIN
;
;
T01010: IST @ASEL2 ; ERROR FLAG SET ?
HGE T01011 ; NO : O.K.
;
;
JSR PC,PESE ; MODEM STATUS
;*****
;ERRNS,BEGIN,TABMOD ;*****
;*****
;MSGSS,AMES06,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
BR T01013 ; END
;
;
T01011: CMP ADDCON,@ASEL4 ; FRAME ?
BNE T01012 ; BAD
CMP DATA12,@ASEL10 ; DATA 1,2 ?
RNE T01012 ; BAD
CMPB DATA3,@ASEL12 ; DATA 3 ?
BEQ T01013 ; GOOD : END
;
;
T01012:
;*****
;ERRNS,BEGIN,TABKMC ;*****
;*****
;MSGSS,AMES07,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
;
;
T01013:
BREAKS,BEGIN ; TEMPORARY RETURN TO MONITOR...
BFEAKS,BEGIN ; CONTINUE AT NEXT INSTRUCTION.
DECB LOOPCN ; LOOP COUNT
BNE T01004 ; # 0 : LOOP
;
;
T01014: INC LINUMB ; NEW LINE NUMBER
CLC ;
ASL ; SHIFT TEST MASK
BCC T01002 ; NOT LAST U.U. , LOOP

```

Address	Instruction	Comment
325	.SWTTL	TEST 2
326		
327		
328		
329		
330		
331		
332		
333		
334		
335		
336		
337		
338		
339		
340		
341		
342		
343	DMT02: MOV	#100, LOOPCN ; INIT LOOP COUNT
344		
345	T02002:	
346	GETPAS, INFO2	; GET PHYSICAL ADDRESS OF INFO2
347	ASL INFO2+4	; ROTATE EXTENSION BITS
348	ASL INFO2+4	
349	SWAB INFO2+4	; TO BITS 15-14
350	MOVE #377, INFO2+4	; MAXIMUM BIT COUNT
351	MOV BAR1, BAR2	; LINE MASK
352	MOV #1, L1N	; SELECT LINE 0
353	CLR L1N	; LINE NUMBER = 0
354	T02003: MOV	#3, DATA3 ; 3 TRANSMISSIONS BY L.U.
355	BIT LIN, BAR2	; IS THE LINE PRESENT ?
356	BNE T02004	; YES : TRY IT
357	JMP T02024	; NO : JUMP
358		
359	T02004: JSR	PC, CLEKMC ; INIT KMC
360		
361	T02005: JSR	PC, ROCSR ; REQUEST REGISTERS
362	GETPAS, ABUFIN	; GET PHYSICAL ADDRESS OF ABUFIN
363	MOV ABUFIN+2, @ASEL10	
364	ASL ABUFIN+4	; ROTATE EXTENSION BITS
365	ASL ABUFIN+4	
366	SWAB ABUFIN+4	; TO BITS 15-14
367	MOV ABUFIN+4, @ASEL12	
368	CLR @ASEL4	; CLEAR SEL4
369	MOV ADDR, R1	; KMC BASE ADDRESS
370	MOV LINUMB, 3(R1)	; LINE NUMBER
371	MOVE #RXJN, (R1)	; RX IN
372	CLR @ASEL2	; CLEAR BSEL2
373	DECR (PC)+	; TEMPO
374		
375	BNE 1S	
376	JSR PC, ROCSR	; REQUEST REGISTERS
377	GWBUFS, @BEGIN	; GET WRITE BUFFER INFORMATION
378	MOV #RUFEEA, INFO1	; WRITE BUFFER ADDRESS
379	MOV #RUFEEA, INFO1+3	
380	ASLF INFO1+3	; ROTATE EXTENSION BITS

DESCRIPTION:
 TRANSMISSION AND RECEPTION IN A
 LOOPING BACK OF A INFORMATION
 FRAME
 INITIALIZATION OF THE LINE UNIT
 SENDING A TABLE ADDRESS FOR RX
 TRANSMISSION OF A RANDOM FRAME (LENGTH AND CONTENT)
 WAITING FOR TWO MESSAGES FROM KMC
 - ACKNOWLEDGE OF THE TRANSMISSION
 - END OF RECEPTION
 CHECKING OF THE CONTENT OF THE TWO FRAMES

```
381 001450 106367 002301 ASLK INFO1+3 ; TO BITS 15-14
382 001454 JSR PC,RANDOM
383 001460 MOVH KANNUM,DATA12
384 001466 BEQ 2$ ; AVOID ZERO-LENGTH FRAME
385 001470 MOVH KANNUM,INFO1+2 ; CHARACTER COUNT
386
387 001476 JSR PC,RANDOM
388 001502 ISIB KANNUM ; AVOID 0 IN AD CHARACTER
389 001506 BEQ 3$ ; YES : ANOTHER ONE
390
391 001510 GETPAS,ABUFOU ;GET PHYSICAL ADDRESS OF ABUFOU
392 001514 MOV ABUFOU+2,PASEL10
393 001522 ASL ABUFOU+4 ; ROTATE EXTENSION BITS
394 001526 ASL ABUFOU+4
395 001532 SWAH ABUFOU+4 ; IO BITS 15-14
396 001536 MOV ABUFOU+4,PASEL12 ; CLEAR BIT 8
397 001544 BIC #400,KANNUM ; LOAD IN SEL4
398 001552 MOV KANNUM,PASEL4 ; SAVE THEM
399 001560 MOV KANNUM,ADDCON ; KMC BASE ADDRESS
400 001566 MOV ADDR,R1 ; LINE NUMBER
401 001572 MOVH LINUMB,3(R1) ; TX INFO FRAME
402 001600 MOVE #TXIN,(R1)
403
404 001604 JSK PC,WTANS ; WAIT ANSWER FROM KMC
405 001610 CMPB #24,PASEL2 ; END OF TX ?
406 001616 BEQ T02014 ; YES
407 001620 CMPB #20,PASEL2 ; END OF RX ?
408 001626 BEQ T02015 ; YES
409
410 001630 JSR PC,MESE ; MODEM STATUS
411 001634 ***** ; *****
ERRNS,BEGIN,TABMOD ; *****
MSGSS,AMES08,BEGIN ; ASCII MESSAGE CALL WITH NO HEADER
BR T02024 ; END
412 001642 ***** ; *****
413 001650 ***** ; *****
414 ***** ; *****
415 001652 ***** ; *****
416 001656 ***** ; *****
417 ***** ; *****
418 001660 ***** ; *****
419 ***** ; *****
EPKNS,BEGIN,TABMOD ; *****
MSGSS,AMES09,BEGIN ; ASCII MESSAGE CALL WITH NO HEADER
BR T02024 ; END
420 001672 ***** ; *****
421 001700 ***** ; *****
422 ***** ; *****
423 001702 ***** ; *****
424 001706 ***** ; *****
425 ***** ; *****
426 001710 ***** ; *****
427 001714 ***** ; *****
ERRNS,BEGIN,TABMOD ; *****
MSGSS,AMES10,BEGIN ; ASCII MESSAGE CALL WITH NO HEADER
BR T02024 ; END
428 001722 ***** ; *****
429 001730 ***** ; *****
```


DMSA DEC/X11 SYSTEM EXERCISFR MACRO M1113 01-JUL-60 16:42 PAGE 12
END OF PASS

461
462
463
464
465

.SBTTL END OF PASS

;
;
;
PASS:

002074
104402 000000

ENDPSS,BEGIN ;SIGNAL END OF PASS, CONTINUE AT RESIRT

.SBTTI. AUTOMATIC LINE CONFIGURATION

LOAD PAR2 WITH ONE 1 BIT FOR EACH LINE PRESENT

INPUTS : ADDR = BASE DEVICE ADDRESS

OUTPUTS : BAR2

CALLING SEQUENCE :

JSR PC,LINAUT

; INIT CSR
; INIT PARAMETERS

; CLEAR SEL0
; MASTER CLEAR
; CLEAR SEL0

; ADDRESS LINE
; LINE NUMBER
; CHOOSE A LINE NUMBER

; LINE TEST

; READ IF ON LINE IN (R1)

; COMPARE WITH EXPECTED VALUE

; NOT HERE, SUPPRESS IT

LINE NO.	ADDRESS	OPERATION	OPERAND	COMMENT
467				
468				
469				
470				
471				
472				
473				
474				
475				
476				
477				
478				
479				
480				
481				
482				
483				
484				
485	002100	MOV	175702	
486	002104	MOV	016701	
487	002112	MOV	012767	
488	002120	MOV	177777	
489	002126	MOV	001624	
490		CLR	001620	
491	002132	CLR	001614	
492	002134	MOV	005067	
493	002140	MOV	005011	
494	002142	MOV	000100	
495	002150	MOV	122010	
496	002156	MOV	001560	
497	002164	MOV	001554	
498	002170	MOV	000006	
499	002176	MOV	000001	
500		CLR	000001	
501	002202	MOV	021200	
502	002210	MOV	012767	
503	002216	MOV	016761	
504	002224	MOV	112761	
505	002232	MOV	000002	
506		CLR	000001	
507	002236	MOV	105061	
508	002242	MOV	111167	
509	002250	MOV	001472	
510	002256	BIS	000020	
511	002264	CPH	056767	
512		BEQ	001470	
513	002266	BIC	126767	
514		LIN, BAR1	001403	
515	002274	INC	001450	
516	002300	CLC	001442	
517	002302	ASL	001444	
518	002306	BCC	000241	
519	002310	RTS	006367	
			103311	
			000207	

LINAUT:

3S:

7S:

PC, LINAUT

ADDR, R1

#-1, BAR1

#-1, BAR2

#1, L1N

LINUMB

(R1)

#100, (R1)

(R1)

#244*400+10, INSTLO

INSTLO, 6(R1)

#2, 1(R1)

LINUMB, (R1)

#3, 1(R1)

1(R1)

#42*400+200, INSTLO

INSTLO, 6(R1)

#2, 1(R1)

#3, 1(R1)

1(R1)

(R1), BAD

#20, GOOD

LINUMB, GOOD

HEAD, GOOD

7S

LIN, BAR1

LINUMB

LIN

3S

BCC

RTS

PC

RTS

PC

PC

PC

PC

PC

PC

PC

PC

PC

PC

KMC CODE LOADER

```

521 .SHTTL KMC CODE LOADER
522
523
524
525
526
527
528
529
530
531
532
533
534 002312 016700 006216
535 002316 016701 175464
536 002322 012702 001777
537
538 002326 105061 000001
539 002332 112761 000004
540 002340 010261 000004
541 002344 014061 000006
542 002350 152761 000040
543 002356 142761 000040
544 002364 021061 000004
545 002370 001003
546
547 002372 005302
548 002374 002354
549
550 002376 000207
551
552 002400
553 002400 104410 000000 004050
554 002406 104416 003400 000000
555 002414 104403 000000

```

DESCRIPTION:

ROUTINE TO LOAD KMC CODE

CALLING SEQUENCE:

JSK PC,LOADER

LOADER: MOV KMCODE,R0 ; LOAD CODE IN KMC CRAM
 MOV ADDR,R1 ; BASE ADDRESS
 MOV #1777,R2

1S: CLR R1 ; SET RAM0 TO ONE
 MOV #4,1(R1) ; LOAD CRAM ADDRESS
 MOV R2,4(R1) ; LOAD MICRO-INSTRUCTION
 MOV -(R0),6(R1) ; CRAM WRITE
 BISR #40,1(R1) ; CLEAR CRAM WRITE
 BICH #40,1(R1) ; CHECK INSTRUCTION
 CMP (R0),4(R1)
 BNE ZS

DFC R2
 BGE 1S

RTS PC

```

2S:
;*****
ERRNS,BEGIN,TABKMC ;
;*****
MSGSS,AMES12,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
ENDS,BEGIN ; BAD KMC LOAD , DROP THE MODULE

```

```

556 ;
557 ;
558 ;
559 ;
560 ;
561 ;
562 ;
563 ;
564 ;
565 ;
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 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 MOVB #200,1(R1) ; START KMC CODE
577 002452 004767 000122 JSR PC,RGCSR ; REQUEST CSR REG
578 002456 104413 004004 GETPAS,ACSREX ;GET PHYSICAL ADDRESS OF ACSREX
579 002462 016777 001320 MOVB 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 MOVB ACSREX+4,@ASEL6 ; IN KMC
584 002512 112777 000007 MOVB #CONM,@ASEL0
585 002520 004767 000164 JSR PC,WIANS ; WAIT ACKNOWLEDGE
586 002524 005777 001322 TST @ASEL2 ; ERROR FLAG SET ?
587 002530 002404 BLT 1$ ; YES : ERROR
588 ;
589 002532 122777 000027 CMPB #27,@ASEL2 ; GOOD ANSWER ?
590 002540 001406 BEQ 2$ ; YES : O.K.
591 ;
592 002542 ;
593 002542 104410 000000 *****
594 002550 104416 003420 ***** ERRNS,BEGIN,TABKMC ;
595 002556 012700 000001 ***** ;ASCII MESSAGE CALL WITH NO HEADER
596 002562 004767 000406 JSR PC,GROU ; SET MASK TO ZERO
597 002566 012700 000015 MOVB #GROC,R0 ; EXECUTE
598 002572 004767 000376 JSR PC,GROU ; SET INIT GROUP COMMAND
599 002576 000207 RTS PC ; EXECUTE
  
```


.SBTTL REQUEST CSR REGISTERS (RDYI INTERRUPT)

```

601
602
603
604
605
606
607
608
609
610
611
612
613
614 002600 005067 001210
615 002604 005067 001114
616 002610 005067 001212
617 002614 105077 001232
618 002620 112777 000300 001222
619
620 002626
    002626 104407 000000
    002632 104407 000000
621 002636 005767 001062
622 002642 001013
623
624 002644 005367 001144
625 002650 001366
626
627 002652 011667 001150
628 002656 104416 003404 000000
629 002664 042777 000300 001156
630
631 002672 000207
632
633
634
635
636 002674 042777 000300 001146
637 002702 005167 001016
638 002706 000002
    
```

DESCRIPTION:

REQUEST CSR REGISTERS
 LOAD ROI IN CSR 0
 WAIT FOR RDYI INTERRUPT (CSR 2)

CALLING SEQUENCE:

```

        JSR    PC,ROCSR
        CLR    CLR
        CLR    INTFLG
        CLR    CALLFC
        CLR    #ASEL2
        MOV    #300,#ASELO
        ; CLEAR LOOP
        ; INIT THE SOFTWARE INTERRUPT FLAG
        ; CLEAR CALL ADDRESS
        ; CLEAR BSEL2
        ; SET ROI & IEI
    
```

ROCSR1:

```

        BREAKS,BEGIN
        BREAKS,BEGIN
        TST   INTFLG
        BNE  ROCSR3
        ; TEMPORARY RETURN TO MONITOR...
        ; CONTINUE AT NEXT INSTRUCTION.
        ; DID THE INTERRUPT OCCURS ?
        ; YES : NORMAL RETURN
    
```

ROCSR3:

```

        DEC    CLR
        BNE  ROCSR1
        MOV    (SP),CALLPC
        MSGSS,AMES13,BEGIN
        BIC   #300,#ASELO
        ; SET CALL ADDRESS
        ; ASCII MESSAGE CALL WITH NO HEADER
        ; CLEAR ROI & IEI
        ; NORMAL RETURN
    
```

INTRDI:

```

        BIC   #300,#ASELO
        COM  INTFLG
        RTI
        ; CLEAR ROI & IEI
        ; SET THE INTERRUPT FLAG
        ; 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 ;
    
```

```

; RANNAD, RANNUM
; #2, RANNAD
; 1S
; CLR RANNAD
; RTS
; CLEAR RANDOM ADDRESS IF > 100000
    
```

```

; RANNAD: 0
; RANNUM: 0
; RANDOM ADDRESS NUMBER
; RANDOM NUMBER VALUE
    
```

SBTTL RANDOM NUMBER GENERATOR SUBROUTINE

703 ;
704 ;
705 ;
706 ;
707 ;
708 ;
709 ;
710 ;
711 ;
712 ;
713 ;
714 ;
715 ;
716 ;
717 ;
718 ;
719 ;
720 ;
721 ;
722 ;
723 ;
724 ;
725 ;
726 ;
727 ;
728 ;
729 ;
730 ;
731 ;
732 ;

.SBTTL ERROR MESSAGE FROM KMC

DESCRIPTION :

SAVE KMC STATUS
REQUEST MODEM STATUS
WAIT FOR KMC ANSWER

CALLING SEQUENCE :

JSR PC,MESE

MOV	000760	017767	003064	017767	000760	001022	MESE:	MOV	@ASEL0,BUFKMC	: SAVE SEL0
MOV	000754	017767	003072	017767	000754	001016		MOV	@ASEL2,BUFKMC+2	: SAVE SEL2
MOV	000750	017767	003100	017767	000750	001012		MOV	@ASEL4,BUFKMC+4	: SAVE SEL4
MOV	000744	017767	003106	017767	000744	001006		MOV	@ASEL6,BUFKMC+6	: SAVE SEL6
MOV	000740	017767	003114	017767	000740	001002		MOV	@ASEL10,BUFKMC+10	: SAVE SEL10
MOV	000734	017767	003122	017767	000734	000776		MOV	@ASEL12,BUFKMC+12	: SAVE SEL12
MOV	000730	017767	003130	017767	000730	000772		MOV	@ASEL14,BUFKMC+14	: SAVE SEL14
MOV	000724	017767	003136	017767	000724	000766		MOV	@ASEL16,BUFKMC+16	: SAVE SEL16
JSR	003144	004767	003144	004767	177430			JSR	PC,R0CSR	: REQUEST CSR REGISTER
MOV	016701	016701	003150	016701	174632			MOV	ADDR,R1	: KMC BASE ADDRESS
MOV	000564	116761	003154	116761	000564	000003		MOV	LINUM,3(R1)	: LINE NUMBER
JSR	000003	112711	003162	112711	000003			JSR	#A0ST,(R1)	: WAIT RDYO
RTS	177516	004767	003166	004767	177516			RTS	PC,WTANS	
	000207	000207	003172	000207					PC	

```
734 .SBTTL SPECIAL COMMANDS
735
736
737
738 DESCRIPTION :
739 SEND SPECIAL COMMAND TO KMC
740 WAIT FOR ACKNOWLEDGE.
741
742
743 CALLING SEQUENCE :
744 JSR PC,GROU
745 WITH COMMAND TYPE IN RO
746
747
748
749
750
751 GROU: MOV K0,TYPCOM ; SAVE COMMAND TYPE
752 JSR PC,ROCSR ; REQUEST CSR REG
753 MOV TYPCOM,BASEL6 ; COMMAND TYPE
754 CLR BASEL4
755 MOV ADDR,R1 ; KMC BASE ADDRESS
756 MOVB #COMM,(R1) ; LINE NUMBER
757 JSR LINUMB,3(R1) ; WAIT ACKNOWLEDGE
758 TST PC,WIANS ; ERROR FLAG SET ?
759 BLT BASEL2 ; YES
760
761
762 CMPB #27,BASEL2 ; GOOD ANSWER ?
763 BEQ 2$ ; YES : RETURN
764
765
766 *****
767 ERRNS,BEGIN,TABKMC ;
768 MSGSS,AMES16,BEGIN ;ASCII MESSAGE CALL WITH NO HEADER
769
770 *****
771 RTS PC
772
773 2$:
774 1$:
```

ASCII STRINGS

770 ;
771 ;
772 ;
773 ;
774 003274 003430' 003455' 177777
775 003302 003430' 003501' 177777
776 003310 003430' 003515' 177777
777 003316 003531' 177777
778 ;
779 003322 003437' 003455' 177777
780 003330 003437' 003501' 177777
781 003336 003437' 003515' 177777
782 003344 003550' 177777
783 ;
784 003350 003446' 003455' 177777
785 003356 003446' 003501' 177777
786 003364 003446' 003515' 177777
787 003372 003446' 003566' 177777
788 ;
789 003400 003603' 177777
790 ;
791 003404 003630' 003650' 177777
792 ;
793 003412 003640' 003650' 177777
794 ;
795 003420 003664' 177777
796 ;
797 003424 003703' 177777

.SBITL ASCII STRINGS
AMES00: MESS00,MESS03,177777
AMES01: MESS00,MESS04,177777
AMES02: MESS00,MESS05,177777
AMES03: MESS06,177777
AMES04: MESS01,MESS03,177777
AMES05: MESS01,MESS04,177777
AMES06: MESS01,MESS05,177777
AMES07: MESS07,177777
AMES08: MESS02,MESS03,177777
AMES09: MESS02,MESS04,177777
AMES10: MESS02,MESS05,177777
AMES11: MESS02,MESS08,177777
AMES12: MESS09,177777
AMES13: MESS10,MESS12,177777
AMES14: MESS11,MESS12,177777
AMES15: MESS13,177777
AMES16: MESS14,177777

ASCII STRINGS

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 /FFAME /
	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	105	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
003706	103	111	101
003711	114	040	103
003714	117	115	115
003717	101	116	104
003722	045	000	

814

MESS14: .ASCIZ /SPECIAL COMMAND%/

.EVEN

PROGRAM VARIABLES

```

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

```

.SHTTL PROGRAM VARIABLES

```

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

```

```

857
858
859
860
861 004030 004114*
862 004032 004116*
863 004034 004120*
864 004036 004122*
865 004040 004124*
866 004042 004126*
867 004044 004130*
868 004046 004132*
869 004050
870 004050 000000
871 004052 000000
872 004054 000000
873 004056 000000
874 004060 004104*
875 004062 004106*
876 004064 004110*
877 004066 004112*
878 004070 004006*
879 004072 004010*
880 004074 004026*
881 004076 004012*
882 004100 003746*
883 004102 177777
884
885
886
887 004104
888 004114
889 004134

```

```

;
;
;
TABMOD: BUFKMC
          BUFKMC+2
          BUFKMC+4
          BUFKMC+6
          BUFKMC+10
          BUFKMC+12
          BUFKMC+14
          BUFKMC+16

TABKMC:
ASEL0:  0
ASEL2:  0
ASEL4:  0
ASEL6:  0
ASEL10: CSREX
ASEL12: CSREX+2
ASEL14: CSREX+4
ASEL16: CSREX+6
          ACSREX+2
          ACSREX+4
          CALLPC
          ADDCON
          DATA12
          177777
;
;
;
CSREX:  .BLKW  4
BUFKMC: .BLKW  10
BUFIN:  .BLKB  400

```

-.SRTTL. BUFFERS

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

.SBTTL *KMC* *** DMS11F KMC MICROCODE ****
.LIST AC
.MLIST MF

KSTART: 10 004534

.IDENT /X01.01/

* DMS11F *

COPYRIGHT (C) 1978 BY
DIGITAL EQUIPMENT FRANCE, ANNECY

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH A LICENSE 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

- RE 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
60
61
62
63
64
65
66
67
68
69
70
71
72

DATE : APRIL 14, 1978

VERSION: 03

HISTORY:
14 APRIL 1978 : CREATION
23 JULY 1979 : ADD ROUTINE FOR GARBAGE RECEIVED DATA WHEN
NO BUFFER IS AVAILABLE
30 OCTOBER 1979 : SUPPRESS ROUTINES FOR ERROR CLASSES
MODIFY RECEPTION BUFFER HANDSHAKING (NO
MORE SUPPRESSION WHEN ERRONEOUS CONTROL
FRAME RECEIVED)
MODIFY SCAN REGISTER HANDSHAKING

KMC DEFINITIONS

```

74 .SHTTL *KMC* DEFINITIONS
75 ;
76 ; *****
77 ; * ASSEMBLER PARAMETERS *
78 ; *****
79 ;
80 ;
81 ;
82 ;
83 ;
84 ;
85 ;
86 ;
87 ;
88 ;
89 ;
90 ;
91 ;
92 ;
93 ;
94 ;
95 ;
96 ;
97 ;
98 ;
99 ;
100 ; *****
101 ; * DEFINITIONS *
102 ; *****
103 ;
104 ;
105 ;
106 ;
107 ;
108 ;
109 ;
110 ;
111 ;
112 ;
113 ;
114 ;
115 ;
116 ;
117 ;
118 ;
119 ;
120 ;
121 ;
122 ;
123 ;
124 ;
125 ;
126 ;
127 ;
128 ;
129 ;
130 ;

; *****
; * DEFINITIONS *
; *****

; MACRO RE-DEFINITION OF LDMAP
;
; MACRO LDMAP SRC,DATA
; <MOVE:LDMAPG:IMM:<DATA>>
;
;
;
; MASK FOR SCAN STATUS
;
; MFII=277 ; FIFO INPUT READY
; MFIOU=337 ; FIFO OUTPUT FREE
; MSTIN=357 ; STATUS INPUT LOADED
; MSTOU=367 ; STATUS OUTPUT READ
;
;
; LINE UNIT REGISTERS DEFINITION
;
; INPUT
;
; RELUAD=200 ; DECODED L.U. ADDRESS
; RESCAN=240 ; SCAN STATUS
; REFICO=300 ; FIFO CONTROL
; REFIDA=320 ; FIFO DATA
; RESTLO=340 ; STATUS LOW

```

```

131 000360 RESTHI=360 ; STATUS HIGH
132 ;
133 ; OUTPUT
134 ;
135 002010 LOLUAD=10:2000 ; L.U. ADDRESS
136 002012 STSCAN=12:2000 ; START SCAN
137 002014 LOFICG=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 SP8=10
146 000011 SP9=11
147 000012 SPI0=12
148 000013 SPI1=13
149 000014 SPI2=14
150 000015 SPI3=15
151 000016 SPI4=16
152 000017 SPI5=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 OVF
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 BEGFR=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 LFDATA1=12 ; LONG FR: DATA 1
180 000011 LFDATA2=11 ; LONG FR: DATA 2
181 000010 LFDATA3=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

```

KMC DEFINITIONS

188	000001	HLEF=1	; HEGN FRAME + EOF
189	000002	WCTRL=2	; WT CONTROL CHARACTER
190	000003	CTRL=3	; CONTROL CHAR + EOF
191	000004	LF1=4	; LONG FRAME + DATA1
192	000005	LF01EF=5	; IDEM + EOF
193	000006	LF2=6	; LONG FRAME + DATA2
194	000007	LF02EF=7	; IDEM + EOF
195	000010	LF3=10	; LONG FRAME + DATA3
196	000011	LF03EF=11	; IDEM + EOF
197	000012	ABOF=12	; ABOFT PECEPTION
198	000013	ABOPEF=13	; ODEM + EOF
199	000014	ABORR=14	; ABOFT REQUESTED BY PDP
200	000015	ABORRF=15	; IDEM + EOF
201	000016	INOR=16	; INFO FRAME, NO BUFFER
202	000017	INORF=17	; INFO, NEW BUF, CHAR 1,EVEN
203	000020	INIE=20	; IDEM = EOF
204	000021	INIEEF=21	; INFO, NEW BUF, CHAR 1,ODD
205	000022	INIO=22	; IDEM,EOF
206	000023	INIOEF=23	; INFO, NEW BUF, CHAR2
207	000024	IN2F=24	; IDEM + EOF
208	000025	IN2EEF=25	; INFO, BODY, CHAR1
209	000026	IE1=26	; IDEM + EOF
210	000027	IE1EF=27	; INFO, BODY, CHAR2
211	000030	IR2=30	; IDEM + EOF
212	000031	IR2EF=31	; INFO, BODY, CHAR2
213	000032	IE1=32	; IDEM + EOF
214	000033	IE1EF=33	; INFO, END BUF, CHAR1
215	000034	IE2=34	; IDEM + EOF
216	000035	IE2EF=35	; INFO, END BUF, CHAR2
217			; IDEM + EOF
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	XMASK=21	; OR MASK
225			
226			
227			
228			
229			
230	000040	BLOCK=40	; BLOCK LENGTH
231	000265	BADA=265	; BAD ADDRESS
232			
233			
234			
235			
236			
237			
238			
239	000020	RDY0=20	; SET RDYO
240	000200	RDY1=200	; SET RDI
241	000200	ITIN=200	; SET INTERRUPT IN
242	000300	ITOUT=300	; SET INTERRUPT OUT
243			
244			

```

245 *KMC* DEFINITIONS
246
247
248
249 000000
250 000002
251 000001
252
253
254
255
256 000000
257 000002
258 000001
259 000003
260 000004
261 000006
262 000005
263 000007
264
265
266
267 000347
268 000337
269 000376
270 000075
271
272 000020
273
274 000377
275 000375
276 000377
277 000306
278
279

```

```

; KMC STATES
;
;
; IDLE =0
; WTIN =2
; WTOUT =1
;
;
; COMMAND TYPE IN AND OUT
;
; RXINFO=0
; RXLLOC=2
; RXSHCO=1
; MODSTA=3
; TXINFO=4
; TXLLOC=6
; TXSHCO=5
; COMMAN=7
;
;
; MASKS FOR ERROR CLASSES
;
;
; MC11=347
; MC21=337
; MC31=376
; MC41=75
;
; MC12=20
;
; MC13=377
; MC23=375
; MC33=377
; MC43=306
;
;
; ERR 1
; ERR 2
; ERR 3

```


KMC INITIALISATION

```

281 .SRTTL *KMC* INITIALISATION
282 ;
283 ;
284 ;
285 ; *****
286 ; * INITIALISATION *
287 ; *****
288 ;
289 ;
290 ; CODE ENTERED WHEN MASTER CLEAR
291 ; IS ISSUED BY PDP11
292 ; MAR IS 0
293 ; HRG IS 0
294 ; PC IS 0
295 ; PDP IS SUPPOSED TO CLEAR CSR
296 ;
297 ;
298 ; CLEAR KMC STATUS (IDLE)
299 ;
300 ;
301 DEBUT: SP BK,SELB,SP7
302 ; CLEAR MAR
303 ;
304 DEBUT1: MEM IMM,0,INCMAR ; CLEAR BYTE
305 SP IPUS,NPR,SPO ; TEST IF MAR
306 BRWRTE RR,TMOA,SPO ; OVER FLOW
307 BP7 SC00
308 ALWAYS DEBUT1 ; LOOP UNTIL END
309 ;
310 ;
311 ;

```

```
*KMC* MAIN LOOP
313 .SBTTL *KMC* MAIN LOOP
314 ;
315 ;
316 ;
317 ;
318 ;
319 ;
320 ;
321 ;
322 ;
323 ;
324 ;
325 ;
326 ;
327 ;
328 ;
329 ;
330 ;
331 ;
332 ;

; THREE TYPES OF REQUEST CAN HAPPEN
; -L.U. REQUEST (IN L.U. SCAN STATUS)
; -PDP REQUEST (IN CSR REGISTERS)
; -OUTPUT TO PDP REQUEST (IN INTERNAL QUEUE)

; THE MAIN LOOP
; SERVICES ONE L.U. (IF REQUEST)
; SERVICES ONE OUTPUT TO PDP OK ONE PDP REQUEST (IF NO OUTPUT
; TO DO)
; THEN IT BEGINS AGAIN
```

```

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 SPI2 AND SPI3
344 ;
345 004550
346 004550
347
348 004552
349 004554
350 004556
351 004500
352
353
354
355
356
357 004562
358 004564
359 004566
360 004570
361 004572
362 004574
363
364
365
366
367 004576
368 004600
369 004602
370
371
372
373
374 004604
375 004606
376 004610
377 004612
378
379
380
381
382
383
384 004614
385 004616
386
387 004620
388 004622
389
390 004624

;SAVE SCAN REGISTER WHEN INTEREST
; (30 OCT 79)
; FIND BLOCK ADDRESS

LU0:
SP BR,SELB,SP6
SP BR,TWOA,SP1
SP BR,TWOA,SP1
SP BR,TWOA,SP1
SP BR,TWOA,SP1,LDMAR
;
; CLEAR DATA COUNT
; TEST IF FIFO INPUT
;
BRWRT IMM,0
SP BR,SELB,SPI4
SP BR,SELB,SPI5
BRWRT IMM,MFIN
NODST BR,AORB,SP6
Z LUFIO
;
; TEST IF FIFO OUTPUT
;
BRWRT IMM,MFIOW
NODST BR,AORB,SP6
Z LUFOW
;
; TEST IF STATUS INPUT
;
BRWRT IMM,MSTIN
NODST BR,AORB,SP6
Z LU30
ALWAYS SC30
; EXIT
;
; STATUS INPUT
; BRANCH DEPENDING UPON CONTENTS OF
; STATUS HIGH BITS 0-2
;
LU30:
SP BR,AANDB,SP6
NODST SELA,SPI,LDMAR
;
SP IBUS,RESTLO,SPI2
SP IRUS,RESTHI,SPI3
; GET STATUS LOW
; GET STATUS HIGH
BRWRT IMM,7

```

KAC SERVICING L.U. REQUESTS

391 004626
 392 004630
 393 004632
 394 004634
 395 004636
 396 004640
 397 004642
 398 004644
 399 004646
 400 004650
 401 004652
 402 004654
 403
 404
 405
 406
 407
 408
 409
 410 004656
 411 004660
 412 004662

LU40:
 ; ALWAYS BR,ADD,SPR,P0 ; ACK
 ; ALWAYS LUACO ; EOF RX
 ; ALWAYS FS00 ;
 ; ALWAYS SC30 ; FIF TR
 ; ALWAYS LUFT0 ; MOD ST
 ; ALWAYS LUMS0 ;
 ; ALWAYS SC30 ;
 ; ALWAYS LUEK0 ; ERROR
 ; ;
 ; ;

; SAVE STATUS USUALLY BEFORE EXITING
 ; FROM A L.U. OR PDP REQUEST
 ; ON TRANSMISSION
 ; ;

SC28:
 ; MODST MEMX,SEIA,SPI,LDMAR
 ; MEMX MEMX,SEIA,SPI0
 ; ALWAYS LU20

000
 001
 010
 011
 100
 101
 110
 111

```

414 *SBT11 *KMC* SCANNING ROUTINE
415 ;
416 ;
417 ;
418 ;
419 ; ***** SCANNING ROUTINE *****
420 ; * SCANNING ROUTINE *
421 ; ***** SCANNING ROUTINE *****
422 ;
423 ;
424 ; SP4 CONTAINS THE LAST PROCESSED OR SCANNED L. U.
425 ; INCREMENT SP4
426 ; LOAD LINE UNIT ADDRESS
427 ; TEST EVENT BIT (SCAN STOP)
428 ; JUMP TO SUCCESSFUL SCANNING ROUTINE IF EVENT BIT SET
429 ; ELSE SERVE PDP' CSR REGISTERS
430 ;
431 ;
432 ;
433 ;
434 ;
435 ;
436 ;
437 ;
438 ;
439 ;
440 ;
441 ;
442 ;
443 ;

```

SC00:

```

SP BR,INCA,SP4
BRWRTI IMM,7
SPBR BK,AANDB,SP4
;
OUT BR,SELB,LOLUAD
SP BR,SELB,SP1
BRWRTI IBUS,RESCAN
BR7 LU0
;
;
;

```

```

; NEW LINE NUMBER
; LOAD LINE UNIT ADDRESS
; LOAD IN SPI AND 1 CLOCK TICK
; READ EVENT BITS AND SAVE THEM
; (30 OCT 79)
; JUMP IF EVENT

```

KMC SERVICING OF CSR REGISTERS

.SBTTL. *KMC* SERVICING OF CSR REGISTERS

445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471 004702
472 004704
473 004706
474 004710
475
476
477
478
479

* SERVICING OF CSR REGISTERS *
* OUTPUT TO PDP OR *
* INPUT FROM PDP *

IT IS STATE DRIVEN ROUTINE
STATE IS IN SP7
THREE POSSIBLE STATES
IDLE: THEN OUTPUT IS ALLOWED
OR IF THE QUEUE IS EMPTY
PDP ROI IS TESTED AND
ACKNOWLEDGED
WT OUTPUT: AN OUTPUT TO PDP IS
GOING ON. WHEN FINISHED,
DO AS IF IDLE STATE
WT INPUT: AN INPUT FROM PDP IS
GOING ON. WHEN FINISHED,
PDP REQUEST IS SERVICED

TEST KMC STATE

BRADDR <SC31-<IDLE*2>>

.ALWAY BR,ADD,SP7,P0

ALWAYS SC40

ALWAYS SC50

SC30:

SC31:

004702
004704
004706
004710

; IDLE
; WT OUTPUT
; FOR WAITING INPUT,

TURN PAGE

```

481 .SETTL *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 ; BRWRTE IBUS, OCON ; GET BSEL2
496 ; SC00 ; INPUT NOT YET
497 ; ; FINISHED,
498 ; ; EXIT
499 ;
500 ; INPUT FINISHED
501 ; SERVICES PDP REQUEST
502 ;
503 ; BLOCK ADDRESS IS LOADED IN MAR
504 ; DEPENDING UPON LINE NUMBER
505 ; LINE NUMBER IS IN BSEL3 BITS 0-2
506 ; SAVE LINE NUMBER IN SP4
507 ;
508 ;
509 P00: SP IBUS, LINENM, SP1
510 SP BR, T*OA, SP1
511 SP BR, T*OA, SP1
512 SP BR, T*OA, SP1
513 SP BR, T*OA, SP1, LDMAR
514 SP IBUS, LINENM, SP4
515 ;
516 ; LOAD LINE NUMBER
517 OUT IBUS, LINENM, LLOAD
518 ;
519 ; NEW STATE = IDLE
520 BRWRTE IMM, IDLE
521 SP BR, SELB, SP7
522 ;
523 ;
524 ; GET TYPE IN RSEL0 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 ; SPBF IBUS, INCUN, SP2 ; READ BSEL0
534 SP BR, SELB, SP0
535 BRWRTE IMM, 7
536 SP BR, AANDR, SP0
537

```

KMC SERVICING OF PDP REQUESTS

```

538 004750
539 004752
540 004754
541 004756
542 004760
543 004762
544 004764
545 004766
546 004770
547
548 004772
549 004774
550 004776
551 005000
552 005002
553
554 005004
555 005006
556 005010
557 005012
558 005014

```

P10:

```

BRADDR P10
  ALWAYS BR,ADD,SP0,P0
  ALWAYS PJ00
  ALWAYS P20
  ALWAYS P20
  ALWAYS PK00
  ALWAYS PB00
  ALWAYS PD00
  ALWAYS PC00
;
SP
  IBUS,PORT3,SP3
  BRWPIE IMM,16
  COMP BR,SP3
Z
C
  BRWPIE IMM,1
  COMP BR,SP3
C
Z
  ALWAYS PA00

```

```

; TABLE AD FOR RX
; INVALID COMMAND
;
; RQ FOR MODEM STATUS
; TX OF INFO FRAME
; TX OF SHORT CONTROL FRAME
; TX OF LONG CONTROL FRAME
;
; COMMAND
;
; ABORT TRANSMISSION (16)
; ABORT RECEPTION (17)
; AND DUMMY COMMAND (20)
;
; MISCELLANEOUS COMMANDS(2 TO 15)
; LOAD MASK (1)
; BASE ADDRESS (0)

```


KMC *T OUTPUT
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575 005016
576 005020
577
578

.SBTIL *KMC* *T OUTPUT
:
:
:
:
:
:
:
:
: KMC STATE IS *T OUTPUT
: IF OUTPUT FINISHED, BRANCH TO
: SERVICING OF OUTPUT TO PDP
: IF NOT, EXIT
:
:
:
:
:
:
: BRWRTE IRUS,OCON ; GET BSEL2
: BR4 SC00 ; NOT FINISHED
: ; FINISHED

```

580 .SRYTL *KMC* SERVICING OF OUTPUT TO PDP
581 ;
582 ;
583 ;
584 ;
585 ;
586 ;
587 ;
588 ;
589 ;
590 ;
591 ;
592 ;
593 ;
594 ;
595 ;
596 ;
597 ;
598 ;
599 ;
600 ;
601 ;
602 ;
603 ;
604 ;
605 ;
606 ;
607 ;
608 ;
609 ;
610 ;
611 ;
612 ;
613 ;
614 ;
615 ;
616 ;
617 ;
618 ;
619 ;
620 ;
621 ;
622 ;
623 ;
624 ;
625 ;
626 ;
627 ;
628 ;
629 ;
630 ;
631 ;
632 ;
633 ;
634 ;
635 ;
636 ;

*****
* SERVICING OF OUTPUT TO PDP *
*****

KMC STATE IS IDLE
; FIRST IF OUTPUT TO BE DONE
; IF YES, OUTPUT MESSAGE AND UPDATE
;   QUEUE
;   NEW KMC STATE IS WT OUTPUT
; IF NO, TEST IF REQUEST FROM PDP
; IF YES, ACKNOWLEDGE REQUEST
;   NEW KMC STATE IS WT INPUT
; IF NO, STAY IDLE

;1-TEST OUTPUT TU PDP
SC40: LDMAP IMM,2 ; POINTER ON LAST
; LDMA IMM,2776 ; MESSAGE
BRWRTI MEMX,SELB,INCMAR
SP ; POINTER ON FIRST
; MESSAGE
COMP BR,SP1 ; TEST IF NO MESS
Z SC42 ; EXIT
;
BRWRTI IMM,10 ; NOW POINTS ON
MEM BR,ADD,SP1 ; MESSAGE TO
; SEND TO PDP
;
; 2 - NPR OUTPUT
; LDMA IMM,2 ; FIND BAD
; LDMA IMM,BADA
;
OUT MEMX,SELB,ORA1,INCMAR
OUT MEMX,SELB,ORA2,INCMAR
SPBR MEMX,SELB,SP0
BRO SC409
; IF NO BAD, NPR IS
; NOT PERFORMED
;
ALWAYS SC415
OUT BR,DECA,ORR
;
NODST BR,SELA,SP1,LDMAR
;
; POINT ON
; MESSAGE
; INIT LOOP
;
SC411: SP IMM,3,SP3,LDMAPG
OUT MEMX,SELB,OUTDA1,INCMAR
OUT MEMX,SELB,OUTDA2,INCMAR
BPWRTI IMM,21
OUT BR,SELB,ONPR
RRWPTI BRBUS,NPR
HPO SC412
SF BR,DECA,SP3
Z SC415
; END, EXIT

```

KMC SERVICING OF OUTPUT TO PDP

```

637 005106          CALLSR  SP5,RXDMA1,SCREL
638                ;
639                ; 3 - CSK OUTPUT
640 005114          BPWRTE  IMM,2,LDMAPG
641 005116          MODST   BF,INCA,SP1,LDMAR
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          BRWRTE  IRUS,INCOM
650 005132          BR4     SC418
651                ;
652 005134          SC416:  BPWRTE  IMM,WJOUT
653 005136          SC417:  SP      BR,SELB,SP7
654                ;
655 005140          ALWAYS  SC00
656                ;
657 005142          BRWRTE  IMM,LTOUT
658 005144          OUT      BR,SELB,ORR
659 005146          ALWAYS  SC416
660                ;
661                ;
662                ;
663                ;
664                ;
665                ;
666                ;
667                ;
668 005150          SC42:  SPAR   IBUS,INCON,SPO
669 005152          BR7     SC43
670                ;
671 005154          BRWRTE  IMM,LDLE
672                ;
673 005156          ALWAYS  SC417
674                ;
675 005160          SC43:  BRWRTE  IMM,0
676 005162          OUT      BR,SELB,OLINEN
677 005164          BRWRTE  IMM,RDYI
678 005166          OUT      BR,SELB,OOCON
679 005170          BRWRTE  BR,TWOA,SPO
680 005172          BR7     SC44
681                ;
682                ;
683 005174          SC430:  BRWRTE  IMM,WJIN
684 005176          ALWAYS  SC417
685                ;
686 005200          SC44:  BRWRTE  IMM,ITIN
687 005202          OUT      BR,SELB,ORR
688 005204          ALWAYS  SC430

```

; INC ADDRESS

; POINT ON
; MESSAGE

; SEE IF INT SET

; NEW KMC
; STATE

; EXIT

; DEAL WITH INT

; GET BSELO

; NO ROI
; STATE=IDLE

; EXIT
; REQUEST
; CLEAR BSEL3 AS
; REQUESTED BY PD
; SET RDI

; SEE IF INT SET

; STATE=WAIT IN
; EXIT
; IT GENERATED

KMC FIFO INPUT

```

690 .SHTTL *KMC* FIFO INPUT
691 ;
692 ; *****
693 ; * FIFO INPUT *
694 ; *****
695 ;
696 ;
697 ; THIS ROUTINE IS ENTERED WHEN THERE IS A LINE UNIT REQUEST B
698 ; BECAUSE OF A RECEPTION
699 ;
700 ; PAGE 1
701 ; LOAD RX STATUS IN SP10 AND TYPE IN IN SP2
702 ; LOAD THE CONTEXT:
703 ; -COUNT
704 ; -CHARACTER ADDRESS
705 ; -DATA SAVED
706 ; GET FIFO CONTROL
707 ; IF END OF FRAME, INCREMENT RX STATUS BY 1 AND ADD BIT
708 ; BIT COUNT IN SP4 ALONG WITH LINE NUMBER
709 ; IF ERROR, SPECIAL HANDLING
710 ;
711 ;
712 ;
713 LUF10: SP BR, AANDB, SP6
714 LDMAP IMM, 1
715 ;
716 SP MEMX, SELB, SP10
717 ;
718 BRWRTE IMM, 6
719 WODST BR, ADD, SP1, LDMAR
720 SP MEMX, SELB, SP2
721 ;
722 BRWRTE IMM, 11
723 WODST BR, ADD, SP1, LDMAR
724 OUT MEMX, SELB, OIAT2, INCMAR
725 SP MEMX, SELB, SPI4, INCMAR
726 SP MEMX, SELB, SPI5, INCMAR
727 OUT MEMX, SELB, ORA1, INCMAR
728 OUT MEMX, SELB, ORA2, INCMAR
729 ;
730 BRWRTE IMM, 100
731 SP IBUS, UBBR, SPO
732 SP BR, AANDB, SPO
733 OUT MEMX, AORB, OBR
734 ;
735 ; BRWRTE IMM, BLOCK
736 ; SP BR, SELB, SP9
737 ;
738 LUF120: SPBR IBUS, REFICO, SP11
739 BR7 LUF150
740 ;
741 ;
742 ;
743 LUF130: BRADDR LUF140
744 .ALWAY BR, ADD, SP10, P0
745 ALWAYS F10000
746 ALWAYS F10100
747 ;
748 ;
749 ;
750 ;

```

```

; UPDATE SCAN STATUS
; PAGE 1
; LOAD STATUS
; LOAD TYPE
; LOAD CONTEXT
; SAVED DATA
; COUNT
; CHAR ADDRESS
; MASK VECT XX4
; HIGH ADDRESS
; INIT BLOCK COUNT
; READ FIFO CONTROL
; BRANCH IF PROBLEM
; FREE
; BEG FRAME + EOF

```

KMC FIFU INPUT

```

751 005264
752 005266
753 005270
754 005272
755 005274
756 005276
757 005300
758 005302
759 005304
760 005306
761 005310
762 005312
763 005314
764 005316
765 005320
766 005322
767 005324
768 005326
769 005330
770 005332
771 005334
772 005336
773 005340
774 005342
775
776
777 005344
778
779 005346
780 005350
781 005352
782 005354
783
784
785
786
787
788 005356
789
790
791
792
793
794
795 005360
796 005362
797 005364
798
799 005366
800 005370
801 005372
802
803 005374
804 005376
805 005400
806 005402

ALWAYS FI0200
ALWAYS FI0300
ALWAYS FI0400
ALWAYS FI0500
ALWAYS FI0600
ALWAYS FI0500
ALWAYS FI0800
ALWAYS FI0900
ALWAYS FI1000
ALWAYS FI1100
ALWAYS FI1000
ALWAYS FI1300
ALWAYS FI1200
ALWAYS FI1400
ALWAYS FI1500
ALWAYS FI1600
ALWAYS FI1700
ALWAYS FI2210
ALWAYS FI1900
ALWAYS FI2000
ALWAYS FI2100
ALWAYS FI2200
ALWAYS FI2300
;
;
LUF150: BR0 LUF152
;
LUF151: SP BR,INCA,SP10
; BRWRTE IMM,34
; SP BR,AANDB,SP11
; ALWAYS LUF130
;
;
; NOW, EKRR CASES
;
;
LUF152: OUT IBUS,REFIDA,OBAl
;
; SET THE FLAG : NO BUFFER AVAILABLE
; WHEN - STATE = INFO FRAME, NO BUFFER
; - STATE = FREE (ERROR ON FIRST CHARACTER)
; - STATE = WAITING CHARACTER ADDRESS (ERROR ON 2ND CHAR)
;
BRWRTE IMM,INOR
COMP BR,SP10
Z LUF153
;
BRWRTE IMM,CTRLEF
COMP BR,SP10
C FI1110
;
LUF153: BRWRTE IMM,17
; MODST BR,ADDV,SP1,LDMAR
; MEM IMM,NOBUFF
; ALWAYS FI1120

; WT CHARACTER ADDRESS
; IDEM + EOF
; LONG FRAME, DATA 1
; IDEM + EOF (ERROR)
; LONG FRAME, DATA 2
; IDEM + EOF (ERROR)
; LONG FRAME, DATA 3
; IDEM +EOF
; ABORT RX
; IDEMM + EOF
; ABORT REQUESTED BY PDP
; IDEM +EOF
; INFO FRAME, NO BUFFER
; IDEMM + EOF
; NEW BUF, CHAR 1, EV
; IDEM + EOF
; NEW BUF, CHAR 1, ODD
; IDEM + EOF
; IDEM + EOF
; CHAR 1
; IDEM + EOF
; CHAR 2
; IDEM + EOF
; ERROR
; END OF FRAME,
; ERROR CASE
; INFO FRAME, NO BUFFER
; 1ST OR 2ND CHAR
; NO BUFFER AVAILABLE

```

```
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 ER,SELR,SP10  
818 005406 ALWAYS LUF162  
819 ;  
820 ;  
821 ;  
822 ;  
823 ;  
824 ; 2 - INFORMATION FRAMES  
825 ;  
826 ; SAVE NEW STATE IN SP10  
827 ; DECREMENT COUNT  
828 ; IF COUNT = ZERO, BRANCH TO FIND NEW BUFFER  
829 ; TEST FIFO : IF NOT READY, SAVE CONTEXT AND EXIT  
830 ; ELSE, GO TO DISPATCHING ROUTINE  
831 ;  
832 ;  
833 ;  
834 005410 BR,SELR,SP10 ; LOAD NEW STATE  
835 005412 BR,DECA,SP14 ; DECREMENT COUNT  
836 005414 LUF168 ;  
837 ;  
838 005416 ;  
839 ;  
840 ;  
841 ;  
842 005416 ; SP ; DEC BLOCK COUNT  
843 005420 BRWRTE BR,TWOA,SP8 ; JUMP IF END  
844 005422 BR7 ; READ FIFO SCAN  
845 ; ; BRANCH FIFO READY  
846 005424 ;  
847 005424 ;  
848 005426 ;  
849 005430 ;  
850 005432 ;  
851 005434 ;  
852 005436 ;  
853 005440 ;  
854 005442 ;  
855 005444 ;  
856 005446 ;  
857 005450 ;  
858 005452 ;  
859 ;  
860 005454 ;  
861 005456 ;  
862 005460 ;  
863 005462 ;  
864 005464 ;
```

LUF157: SP ER,SELR,SP10
ALWAYS LUF162

2 - INFORMATION FRAMES

SAVE NEW STATE IN SP10
DECREMENT COUNT
IF COUNT = ZERO, BRANCH TO FIND NEW BUFFER
TEST FIFO : IF NOT READY, SAVE CONTEXT AND EXIT
ELSE, GO TO DISPATCHING ROUTINE

LUF160: SP BR,SELR,SP10 ; LOAD NEW STATE
LUF161: SP BR,DECA,SP14 ; DECREMENT COUNT
LUF168 ;

LUF163: ; SP ; DEC BLOCK COUNT
; 2 ; JUMP IF END

LUF162: SP ; READ FIFO SCAN
BRWRTE BR,TWOA,SP8 ; BRANCH FIFO READY
BR7 ;

LUF164: ;
NODST BR,SELA,SP1,LDMAR ; NOT READY, SAVE
MEM MEMX,SELA,SP10 ; CONTEXT
BRWRTE IMM,11 ;
NODST BR,ADD,SP1,LDMAR ;
MEM IEUS,INDAT2,INCMAR ;
MEM MEMX,SELA,SP14,INCMAR ;
MEM MEMX,SELA,SP15,INCMAR ;
MEM IEUS,IORAI,INCMAR ;
MEM IEUS,IORAI,INCMAR ;
SP IEUS,IORAI,INCMAR ;
MEM IMM,14 ;
MEM MEMX,AAANDB,SP0,INCMAR ;

LUF165: BRWRTE IEUS,MNPR ; WAIT DMA IF ANY RUNNING
BRU LUF165 ;
BRWRTE IEUS,UBBR ; NON EX MEM
RPO FI1540 ; EXIT
ALWAYS LUI0 ;

```
*KMC*      FIFO INPUT  
865  
866  
867  
868 005466  
869 005470  
870 005472  
871  
872 005474  
873 005476  
874 005500  
875 005502  
  
; DECREMENT COUNT  
LUF168: SP      BR,DECA,SP15  
      Z      LUF169  
      ALWAYS  LUF163  
;  
LUF169: BRWPT  IMM,I2  
      COMP    RR,SP10  
      Z      F12400  
      ALWAYS  F12410  
  
; COUNT HIGH  
  
;LAST CHAR  
; OF THE BUFFER  
; DMA TO PERFORM  
; NO DMA
```

KMC ASK FOR STATUS

```

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 PK00: BRWTE IRUS,RELOAD ; CHECK IF ON LINE
891 005504 BR4 PK01 ; ON LINE
892 005506 ALWAYS TXIN11 ; OFF LINE , HORROR
893 005510 BRWTF IMM,4 ; STATUS DEMAND
894 005514 OUT BR,SELB,LOSTLO
895 005516 BRWTE IMM,24
896 005520 OUT BR,SELB,LOSTHI
897 005522 ALWAYS SC30 ; EXIT

```



```

*AMC* TABLE ADDRESS FOR RECEPTION
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914 005524
915 005526
916
917 005530
918 005532
919 005534
920 005536
921
922 005540
923 005546
924 005554
925 005556
926

*SETTL *AMC* TABLE ADDRESS FOR RECEPTION
;
; *****
; * TABLE ADDRESS FOR RECEPTION *
; *****
;
; PAGE 1
; LOAD RX STATUS IN SP10
; TABLE ADDRESS IS IN BSEL4,5,7
; CHECKS VALIDITY OF REQUEST
; SAVE TABLE ADDRESS IN MAR
; SET TABLE ADDRESS FLAG IN MAR+3
;
; EXIT
;
PJ00:
BRWTE IMM,1
SP BR,SELB,SP0,LDMAPG
;
SP MEMX,SELR,SP10
BRWTE IMM,INOEFF
COMP BR,SP10
C TXIN12
;
CALLSB SPS,BADIN
CALLSB SPS,HIPAAD
MEM MEMX,IMCA,SP8
ALWAYS SC30
;
; RX STATUS
; BUSY, EXIT
; READ BAD

```

```

928 *KMC* INVALID COMMAND
929
930
931
932
933
934
935
936
937 005560
938 005562
939

```

P20:

```

:SBTTL *KMC* INVALID COMMAND
:
: *****
: * INVALID COMMAND *
: *****
:
: QUEUE AN ERROR MESSAGE
:
:
: BPRITE IMM,INVCOM
: ALWAYS EFKMC
:
:

```

KMC TX OF INFO FRAME

```

941 .SHTIL *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 ; CALLSB SP5,TXINI
971 ; CALLSB SP5,BADIN
972 ; CALLSB SP5,HIPAAD
973 ; MEM MEMX,INCA,SP8
974 ;
975 ; BRWRTE IMM,BEGFRA
976 ;
977 ; PB00: SP BR,SELB,SP10,INCMAR
978 ; MEM IMM,0,INCMAR
979 ; MEM INM,0,INCMAR
980 ;
981 ; BRWRTE IMM,BOF
982 ; OUT BR,SELB,LOFICO
983 ; MEM IMM,57
984 ; MEM MEMX,AAADB,SP2,INCMAR
985 ;
986 ; BRWRTE IMM,40
987 ; BRWRTE BK,AAADB,SP2
988 ; BRROTAT
989 ; MEM BR,SELB,INCMAR
990 ;
991 ; MEM IBUS,PORT1,INCMAR
992 ; MEM IBUS,PORT2
993 ; ALWAYS TX00
994 ;
995 ;
; NEW STATE ;
; OVERALL ;
; COUNT TO 0 ;
; BEGIN OF FRAME IN FIFO CTRL ;
; SAVE TYPE IN + FIFOE+PS, IN MAR ;
; GET FIFOE ONLY ;
; PREPARE FOR LAST FIFO CONTROL ;
; LOAD IN MAR ;
; FIRST CHAR ;
; 2ND CHAR ;

```

```
997 ;  
998 ;  
999 ;  
1000 ;  
1001 ;  
1002 ;  
1003 ;  
1004 ;  
1005 ;  
1006 ;  
1007 ;  
1008 ;  
1009 ;  
1010 ;  
1011 ;  
1012 ;  
1013 005646  
1014 005650  
1015 ;  
1016 005652  
1017 005654  
1018 ;  
1019 005656  
1020 005660  
1021 005662  
1022 005664
```

```
*****  
* TRANSMISSION OF INFORMATION FRAME *  
*****  
; 2 - SEND ADDRESS CHARACTERS  
; -----  
; THIS ROUTINE IS ENTERED WHEN TX STATUS IS BEGIN OF FRAME  
; SEND TO FIFO ADDRESS CHAR (MAR + 9)  
; LOAD IN NPR3, CONTROL CHAR  
; LOAD BODY OF FRAME IN FIFO CTRL  
; INIT BLOCK COUNTER IN SP9  
; ;  
; PBA00: BRWRTE IMM,10 ; MAR ON DATA  
; MODST BR,ADD,SP1,LDMAR ;  
; ;  
; OUT MEMX,SELB,LOFIDA,JNCMAR ; FIFO DATA  
; OUT MEMX,SELB,OUTDA2 ;  
; ;  
; BRWRTE IMM,0  
; OUT BK,SELB,LOFICO ; BODY OF FRAME  
; BRWRTE IMM,BLOCK ;  
; SP BR,SELB,SP9 ; BLOCK COUNT
```

KMC TX OF INFO FRAME

```

1024 ;
1025 ;
1026 ; *****
1027 ; * TRANSMISSION OF INFORMATION FRAME *
1028 ; *****
1029 ;
1030 ;
1031 ; 3 - FIRST DMA TO GET CHARACTER ADDRESS
1032 ; -----
1033 ;
1034 ; CALL ROUTINE TO EXECUTE DMA
1035 ; TEMPORARILY SAVE CHARACTER ADDRESS
1036 ; LOW AND MEDIUM IN SPO AND NPR7
1037 ; NEW STATE IS COMPUTED DEPENDING UPON
1038 ; PARITY OF ADDRESS LOW
1039 ;
1040 ; PBA10: BRADDR PBD98 ; ERROR AD
1041 ; SP BR,SELB,SP11
1042 ; CALLSR SP8,TABAD,TABAR1 ;DMA
1043 ; OUT IHUS,INDAT2,OBAA2 ; SAVE DATA
1044 ; SPBR IHUS,INDAT1,SP0
1045 ; BR0 PRA20
1046 ; BRWRTE IMM,EICAR ; NEW STATE
1047 ; ALWAYS PRA30 ; IF EVEN ADDR
1048 ; SP BR,DECA,SP0 ; MAKE ADDRESS EVEN
1049 ; BRWRTE IMM,BEBOUD ; IF ODD ADDR
1050 ; PRA30: SP BR,SELB,SP10

```

```
1052 ;
1053 ;
1054 ;
1055 ;
1056 ;
1057 ;
1058 ;
1059 ;
1060 ;
1061 ;
1062 ;
1063 ;
1064 ;
1065 ;
1066 ;
1067 ;
1068 ;
1069 ;
1070 ;
1071 ;
1072 ;
1073 ;
1074 ;
1075 ;
1076 ;
1077 ;
1078 ;
1079 005720
1080 005726
1081 005730
1082 005732
1083 005734
1084 005736

*****
* TRANSMISSION OF INFORMATION FRAME *
*****

4 - SECOND DMA TO GET CHARACTER COUNT
-----
CALL ROUTINE THAT DOES :
CALL ROUTINE TO PERFORM DMA
THIS ALLOWS TO GET COUNT LOW
COUNT HIGH
LAST BUFFER FLAG
CHAR ADDRESS HIGH
COUNT LOW AND HIGH IN SP14 AND 15
IF LAST BUFFER, HIT COUNT IN MAR + 7
CHAK ADDRESS HIGH IN BRG

LOAD ADDRESS HIGH IN NPR2
LOAD SAVED ADDRESS LOW AND MEDIUM IN
NPR4 AND NPR5

NOW, THE NORMAL TRANSFER OF DATA FROM
PDP MEMORY TO LINE UNIT CAN START

CALLSB SP5,DKAC
MEM MEMX,AORB,SP3
BRWRTE BR,INCA,SP8
OUT BR,SELB,OUTDAI
OUT IBUS,I0BA2,IBA2
OUT SELA,IBA1 ; ADDR HIGH
; ADDR MEDIUM
; ADDR LOW
```

```

1086 ;
1087 ;
1088 ;
1089 ;
1090 ;
1091 ;
1092 ;
1093 ;
1094 ;
1095 ;
1096 ;
1097 ;
1098 ;
1099 ;
1100 ;
1101 ;
1102 ;
1103 ;
1104 ;
1105 ;
1106 ;
1107 ;
1108 ;
1109 ;
1110 ;
1111 ;
1112 ;
1113 ;
1114 ;
1115 ;
1116 ;
1117 ;
1118 ;
1119 ;
1120 ;
1121 ;
1122 ;
1123 ;
1124 ;
1125 ;
1126 ;
1127 ;
1128 005740
1129 005742
1130 005744
1131 005746
1132 005750
1133
1134 005752
1135 005754
1136 005756
1137 005760
1138 005762
1139 005764
1140 005766
1141 005770
1142

*****
* TRANSMISSION OF INFORMATION FRAME *
*****

5 - DATA TRANSFER TO L.U.
-----
REQUEST 2 DATA THRU DMA
EXIT, IF FIFO NOT FREE
EXIT, IF END OF BLOCK
REQUEST NEW BUFFER IF END OF BUFFER
OR SEND END OF FRAME TO FIFO CONTROL
IF LAST BUFFER. IN THIS CASE, ALSO
DEALS WITH FIFO
SEND PREVIOUSLY SAVED DATA TO FIFO

THEN IT IS STATE DRIVEN
STATE: BEGIN OF BUFFER ODD
NEW STATE = 1ST CHARACTER
WAIT FOR END OF DMA
TEST MEMORY INEXISTENT
SAVE DATA HIGH
INCREMENT CHARACTER ADDRESS
GO TO TRANSFER DMA

STATE: 1ST CHARACTER
NEW STATE = 2ND CHARACTER
WAIT FOR END OF DMA
TEST MEMORY INEXISTENT
SAVE DATA LOW
GO TO TEST FIFO FREE

STATE: 2ND CHARACTER
NEW STATE = 1ST CHARACTER
SAVE DATA HIGH
INCREMENT CHARACTER ADDRESS
GO TO TRANSFER DMA

PBC00: OUT IBUS,10DAT1,0NPR ; START DMA
PBC10: BRWRT IBUS,RESCAN ; FIFO FREE?
      BROTAT
      BR4
      PBD00
      INCMA
      IBUS,10DAT2,INCMAR ; START DMA
      IBUS,INDAT2,INCMAR ; FIFO FREE?
      MEMX,SELA,SPI4,INCMAR ; FIFO NOT FREE
      MEMX,SELA,SPI5,INCMAR ; SAVE CONTEXT
      IBUS,I1PA1,INCMAR ; DATA HIGH
      IBUS,I1BA2,INCMAR ; COUNT
      ALWAYS SC28 ; CHAR ADDR
      ; TX STATUS
      ; AND EXIT

```

KMC TX OF INFO FRAME

```

1143
1144 005772 ; BR,DECA,SP9 ; DEC BLOCK
1145 005774 Z PFC20 ; COUNT, TEST
1146 005776 SP BR,DECA,SP14 ; DEC LOW COUNT
1147 006000 Z PBD80
1148 ;
1149 ;
1150 006002 ; IHUS,IODAT2,LOFIDA ; OUTPUT DATA
1151 006004 <PBD20-<HEBUOD*2>> ; TEST STATUS
1152 006006 BR,ADD,SP10,PI
1153 006010 ALWAYS PBD40 ; ODD BUFFER
1154 006012 ALWAYS PBD50 ; 2ND CHAR
1155 ; ; 1ST CHAR
1156 ;
1157 006014 SP BR,DECA,SP10 ; NEW STATE
1158 006016 BRWRTE IBUS,NPR ; WAIT DMA
1159 006020 BR0 PBD30
1160 006022 BRWRTE IBUS,UBER ; MEM INEX?
1161 006024 BR0 PBD98
1162 006026 QUT IRUS,INDAT1,OUTDA2 ; SAVE LOW DATA
1163 006030 ALWAYS PBC10
1164 ;
1165 ;
1166 006032 BRWRTE IRUS,NPR ; ODD BUFFER
1167 006034 BR0 PBD40 ; WAIT DMA
1168 006036 BRWRTE IBUS,UBER ; MEM INEX?
1169 006040 BR0 PBD98
1170 ;
1171 006042 BRWRTE IMM,FICHR ; 2ND CHAR
1172 006044 SP ER,SELB,SP10 ; NEW STATE
1173 006046 QUT IBUS,INDAT2,OUTDA2 ; SAVE HIGH DATA
1174 006050 SP IBUS,IBAI,SP0 ; INC ADDRESS
1175 006052 SP BR,INCA,SP0 ; LOW
1176 006054 QUT BR,INCA,IBAI ; INC ADD HIGH
1177 006056 C PBD60
1178 006060 ALWAYS PBC00
1179 ;
1180 006062 SP IRUS,IBR2,SP0 ; INC AD MEDIA
1181 006064 OUT BR,APLUSC,IBA2
1182 006066 C PBD70
1183 006070 ALWAYS PBC00
1184 ;
1185 006072 BRWRTE IMM,4 ; INC AD HIGH
1186 006074 SP IBUS,IODAT1,SP0
1187 006076 OUT BR,ADD,OUTDA1
1188 006100 ALWAYS PRC00
1189 ;
1190 ;
1191 ;
1192 006102 SP BR,DECA,SP15 ; DEC HIGH COU
1193 006104 Z PBD90
1194 006106 ALWAYS PBD10
1195 ;
1196 006110 BRWRTE MEMX,SELB ; END OF BUFFER
1197 006112 BR7 PRD95 ; LAST BUFFER?
1198 006114 ALWAYS FRA10 ; GET NEW BUFFER
1199 ;

```


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

```

;
;
PBD95:  OUT      BR,SELB,LOFICO
        BR4     1BUS,10DAT2,LOFIDA
        ALWAYS  SC28
        ;      ONMES
;
PBD98:  BR4RTF  1M,TXADM
        ALWAYS  TX000
;
; LAST BUFFER
; LOAD FIFO CO
; OUTPUT DATA
; FIFO, EXIT
; MESSAGE OK
; ERROR
```

```
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 TX000: SP BR,SELB,SP10 ; LOAD NEW STATE
1229 006134 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 *RMC* TX OF LONG CONTROL FRAME
1244
1245 *SBITL *RMC* TX OF LONG CONTROL FRAME
1246 *****
1247 * TX OF LONG CONTROL FRAME *
1248 *****
1249
1250
1251 1 - INITIALISATION
1252 -----
1253
1254 CALL ROUTINE THAT DOES
1255 -LOAD PAGE 0
1256 -CHECKS VALIDITY OF REQUEST
1257 CALL ROUTINE THAT DOES
1258 -READS BAD
1259 -SAVES TWO FIRST CHAR IN MAR
1260 SAVE LAST CHAR IN MAR + 3
1261
1262 NEW STATE IS LONG FRAME, ADDRESS
1263
1264 BRANCH TO END OF INIT ROUTINE
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274

```

PC00:

```

CALLSB SP5,TXINI
CALLSB SP5,BADIN
MEM IRUS,INDATI
BR*RTI IMM,LFADDR
ALWAYS PB0010

```

```

1276 ;
1277 ;
1278 ;
1279 ;
1280 ;
1281 ;
1282 ;
1283 ;
1284 ;
1285 ;
1286 ;
1287 ;
1288 ;
1289 006170
1290 006172
1291 ;
1292 006174
1293 006176
1294 006200
1295 ;
1296 006202
1297 006204
1298 ;
1299 ;

*****
* TX OF LONG CONTROL FRAME *
*****

; 2 - SEND ADDRESS CHARACTER
; -----
;
; THIS ROUTINE IS ENTERED WHEN STATUS IS LONG FRAME, ADDRESS
; SEND ADDRESS CHARACTER TO FIFO DATA
; LOAD BODY OF FRAME IN FIFO CTRL
;
;
PCA00: BRWRTI IMM,10 ; INIT MAR
        NOUPT BR,ADD,SP1,LDMAR
;
        OUT MEMX,SELB,LOFIDA ; FIFO DATA
BRWRTI IMM,0
        OUT BR,SELB,LOFICO ; FIFO CTRL
;
BRWRTI IMM,LFCTRL
        ALWAYS TX000
;
;

```

1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1310
 1311
 1312
 1313
 1314 006206
 1315 006210
 1316 006212
 1317 006214
 1318 006216
 1319 006220
 1320
 1321

```

*****
* TX OF LONG CONTROL FRAME *
*****
    
```

```

3 - OUTPUT CONTROL CHARACTER
-----
    
```

```

; INITIATE MAR TO BLOCK + 9
; OUTPUT DATA
; GO TO DISPATCHING ROUTINE
    
```

```

PCB00:
BWRITE IMM,11
NODST BR,ADD,SPI,LDMAR
OUT MEMX,SELB,LOFIDA
NODST BR,INCA,SPI,LDMAR
BWRITE IMM,LFDAIL
ALWAYS TX000
;
;
    
```

```

; INIT MAR
; OUTPUT DATA
; INIT MAR
; NEW STATE
    
```

KMC TX OF LONG CONTROL FRAME

1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337 006222
1338 006224
1339 006226

```
;  
; *****  
; * TX OF LONG CONTROL FRAME *  
; *****  
; 4 - OUIPUT DATA 1 OR 2  
; -----  
; INCREASE MAR  
; SEND OUT DATA TO FIFO  
; DECREMENT STATE  
; GO TO DISPATCHING ROUTINE  
;  
; PCC00: BRWRIE BR,DECA,SP10 ; DEC TX STATUS  
; OUT MEMX,SELB,LOFIDA ; OUTPUT DATA  
; ALWAYS TX000
```

1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358 006230
1359 006232
1360 006234
1361 006236
1362 006240
1363 006242
1364 006244
1365 006246
1366
1367

* TX OF LONG CONTROL FRAME *

5 - SEND LAST CHARACTER

THIS ROUTINE IS ENTERED WHEN
TX STATUS IS LONG FRAME, DATA 3
LOAD LAST FIFO CONTROL CHARACTER
OUTPUT DATA 3
DEPENDING UPON FIFOE, EXIT STRAIGHT
OR SEND A MESSAGE TO PDP

PCD00: BRWRTE IMM,3 ; MAR ON DATA 3
WDST BR,ADD,SP1,LDMAR ; SAVE DATA 3
OUT MEMX,SELB,OUTDA2,INCMAR
BRWRTE IMM,EOF,INCMAR
SP BR,SELB,SPO,INCMAR
INCMVA ; POINT ON FIFOE
BRWRTE MEMX,ADRB,SPO ; END OF FRAME
ALWAYS PED95
;

1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388 006250
1389
1390 006256
1391 006260
1392
1393 006262
1394
1395 006264
1396
1397

```

.SBTTL *KMC* 1X OF SHORT CONTROL FRAME
;
; *****
; * 1X OF SHORT CONTROL FRAME *
; *****
;
; 1 - INITIALISATION
; -----
; CALL ROUTINE THAT DOES
; -LOADS PAGE 0
; -CHECKS VALIDITY OF REQUEST
; POINTS PAR ON MAR + 3
; NEW STATE IS SHORT FRAME, ADDR
; BRANCH TO END OF INIT ROUTINE
;
; PD00: CALLSR SPS, TXINI
; BRWRTE IMM, 3
; NOUST EP, ADDR, SFI, LDMAR
; BRWRTE IMM, SFADDR
; ALWAYS FB0010
; ;
; ; NEW STATE

```


1399 *****
1400 * TX OF SHORT CONTROL FRAME *
1401 *****
1402 *****
1403 *****
1404 *****
1405 *****
1406 *****
1407 *****
1408 *****
1409 *****
1410 *****
1411 *****
1412 *****
1413 *****

2 - SEND ADDRESS CHARACTER

SEND OUT ADDRESS CHARACTER
NEW STATE = SHORT FRAME, CTRL

1414 PDA00: BRWRTE IMM,10 ; FIFO DATA
1415 006266 BR,ADD,SP1,LDMAR ; NEW STATE
1416 006270 MODST ; EXIT
1417 OUT MEMX,SELS,LOFIDA ;
1418 006272 ;
1419 BRWRTE IMM,SFCTRL ;
1420 006274 ALWAYS TX000 ;
1421 006276 ;
1422 ;
1423 ;

1425
 1426
 1427
 1428
 1429
 1430
 1431
 1432
 1433
 1434
 1435
 1436
 1437
 1438
 1439
 1440
 1441
 1442
 1443 006300
 1444 006302
 1445 006304
 1446 006306
 1447 006310
 1448 006312
 1449 006314
 1450 006316
 1451 006320
 1452
 1453

```
*****
* TX OF SHORT CONTROL FRAME *
*****
```

```
3 - SEND LAST CHARACTER
-----
```

```
THIS ROUTINE IS ENTERED WHEN
TX STATUS IS SHORT FRAME, CTRL
```

```
POINT MAR ON RLOCK + 7
LOAD FIFO CTRL
OUTPUT DATA
DEPENDING UPON FIFOE, EXIT STRAIGHT
OR SEND A MESSAGE TO PDP
```

FDB00:

```
BRWTE IMM,7 ; COMPUTE MAR
WODST BR,ADD,SPI,LDMAR ; GEI FIFOE
BRWTE IMM,EOF
SP BR,SELB,SPO
BRWTE APFX,AORB,SPO,INCMAR ; OUTPUT CTRL
OUT BR,SELB,LOFICO,INCMAR ; OUTPUT DATA
BP4 MEMX,SELB,LOFIDA ; FIFOE, EXIT
ALWAYS LU20 ; END MESSAGE
;
```

KMC LCAD BASE ADDRESS

```

1455 .SBTTL *KMC*   LOAD BASE ADDRESS
1456 *
1457 *
1458 * *****
1459 * * LOAD BASE ADDRESS *
1460 * *****
1461 *
1462 *
1463 * THE THREE BYTES OF BASE ADDRESS
1464 * ARE SAVED IN MAR
1465 * THE COMMENTS OF BASE ADDRESS IS
1466 * READ TO CHECK IF MEMORY EXISTENT
1467 *
1468 *
1469 *
1470 *
1471 *
1472 *
1473 *
1474 *
1475 *
1476 *
1477 *
1478 *
1479 *
1480 *
1481 *
1482 *
1483 *
1484 *
1485 *
1486 *
1487 *
1488 *
1489 *

```

```

;BAD ADDRESS
; LOW
; MEDIUM
; HIGH
; RO DMA
; EXISTENT
; MEMORY?
; OK MESSAGE

```

```

LDMA  IMM,BADA
MEM   IRUS,PORT1
OUT   MEMX,SELB,IRAI,INCMAR
MEM   IRUS,PORT2
OUT   MEMX,SELB,IRAI,INCMAR
SP    IRUS,PORT4,SP8
CALLSB SP5,HIPAI0
MEM   IMM,0
SPBR  BR,INCA,SP8
;
;
OUT   BR,SELB,ONPR
BRWRTE IRUS,NPR
BRU    PA001
BRWRTE IRUS,UBRR
BRU    NOBAD
;
;
MEM   MEMX,SELA,SP8
ALWAYS OKMESS
;
;

```

```

PA00:
PA001:

```

KMC LOAD MASK

```

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 STTUS 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 PE00: CALISE SP5,IXINI ; TEST IF ON LINE
1511 ;
1512 ; OUT IEUS,PORT2,LOSTLO ; SEND AND MASK
1513 BRARTE IMM,AMASK
1514 OUT BR,SELB,LOSTHI
1515 BRARTE IPUS,RESCAN ; WAIT ANSWER
1516 BR4
1517 ALWAYS PE10
1518 ;
1519 PE20: MOUTI IRUS,RESIHI ; ASSUME ACK
1520 OUT IRUS,PORT1,LOSTLO ; SEND OR MASK
1521 BRARIE IMM,XMASK
1522 OUT BR,SELB,LOSTHI
1523 ; ALWAYS SC30
1524 ;
1525 ;

```

```
1527 ; SBTTL *KMC* APOKI TRANSMISSION
1528 ; *****
1529 ; * APOKI TRANSMISSION *
1530 ; *****
1531 ; *****
1532 ; *****
1533 ; *****
1534 ; *****
1535 ; *****
1536 ; *****
1537 ; *****
1538 ; *****
1539 ; *****
1540 ; *****
1541 ; *****
1542 ; *****
1543 ; *****
1544 ; *****
1545 ; *****
1546 ; *****
1547 ; *****
1548 ; *****
1549 ; *****
1550 ; *****
1551 ; *****
1552 ; *****
1553 ; *****
1554 ; *****
1555 ; *****
1556 ; *****
1557 ; *****
1558 ; *****
1559 ; *****
1560 ; *****
1561 ; *****
1562 ; *****
1563 ; *****
1564 ; *****
1565 ; *****

; THIS ROUTINE IS ENTERED WHEN AN APOKI IS REQUESTED
; BY THE PDP

; CHECK IF A TRANSMISSION IS GOING ON
; IF NOT, EXIT
; NEW STATE IS APOKI
; EXIT

; PAGE 0
; GET TX STATE
; FREE, EXIT

; NEW STATE
; EXIT

; SEND APOKI TO FIFO CONTROL
; SEND ANYTHING TO FIFO DATA TO PULL THE QUEUE
; SEND ERROR MESSAGE TO PDP

PF00: SP IMM,0,SPU,LDNAPG ; PAGE 0
; NODST MEMX,SUB,SPO ; GET TX STATE
; C SC30 ; FREE, EXIT
; MEM IMM,IXABOR ; NEW STATE
; ALWAYS SC30 ; EXIT

; *****
; * APOKI TRANSMISSION AFTER REQUEST *
; *****

PFA00: BRWRITE IMM,ABORT
; ALWAYS PFB10
```

1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581 006442
1582 006444
1583 006446
1584 006450
1585 006452
1586 006454
1587
1588 006456
1589
1590

```
;  
;  
;  
;  
;  
; *****  
; * ABORT TRANSMISSION AFTER NON-EXISTENT MEMORY *  
; *****  
; SEND ABORT TO FIFO CONTROL  
; SEND ANYTHING TO FIFO DATA TO PULL THE QUEUE  
; SEND ERROR MESSAGE TO PDP  
;  
;  
;  
;  
; BPWRTE IMM,NOEXME ;  
; MEM IMM,SEOF!BOF> ; OUTPUT ABORT  
; OUT MEMX,SELB,LOFICO ;  
; OUT BR,SELB,LOFIDA ;  
; WODST BR,SELA,SPI,LDMAR ; NEW STATE = FREE  
; MEM IMM,KFREE ;  
; ALWAYS ERKACA ; ERROR MESSAGE  
; ;  
; ;
```

KMC AFOPI RECEPTION

1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610 006460
1611 006462
1612 006464
1613 006466
1614
1615 006470
1616 006472
1617 006474
1618 006476
1619 006500
1620 006502
1621 006504
1622 006506
1623 006510
1624
1625
1626
1627

*SMTL *KMC* ABORT RECEPTION

* ABORT RECEPTION *

THIS ROUTINE IS ENTERED WHEN REQUESTED BY CUSTOMER
EXIT IF DUMMY COMMAND
ELSE,
CHECK IF STATUS IS FREE
IF NOT, NEW STATE IS ABORT REQUESTED
CLEAR TAB ADDRESS FLAG
QUEUE ERROR MESSAGE IN ALL CASES

PG00:

LJMAP IMM,1
BRWRTL IMM,20
COMP BR,SP3
Z SC00

; TEST IF DUMMY COMMAND
; EXIT IF DUMMY

PG10:

SP IMM,0,SPO
COMP MEMX,SPO
Z PG10
MEM IMM,ARERR
BRARTE JEM,BAROLO,INCMAR
INCMA
MEM IMM,0
ALWAYS ERKMC

; TEST IF FREE
; NEW STATE = ABORT REQUESTED
; ERROR = ABORT REQUESTED
; CLEAR TAB ADDRESS FLAG
; SEND ERROR

RMC MISCELLANEOUS COMMANDS

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 *RMC* MISCELLANEOUS COMMANDS

* MISCELLANEOUS COMMANDS *

; TEST IF UNIT ON LINE
; SEND COMMAND IN STATUS HIGH
; ;
; ;

PH00: BRANTE IRUS,RELOAD ; TEST IF ON LINE
BR4 PH01 ; ON LINE
ALWAYS TXINI1 ; OFF LINE, ERROR MESSAGE
PH01: OUT IRUS,PORT3,LOSTHI
ALWAYS SC30

KMC RECEPTION ROUTINES
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662 006524
1663 006526
1664
1670
1671
1672
1679
1680 006530
1681 006530
1682 006532
1683
1684 006534
1685 006536
1686
1687 006540
1688 006542
1689 006544
1690 006546
1691 006550
1692
1693 006552
1694
1698 006554
1700
1701 006556
1702
1703
1704

SA111 *KMC* RECEPTION ROUTINES

* FREE *

SET OVEFALL COUNT TO ZERO
NEW STATE = XT FOR CONTROL CHARACTER
SAVE FIFO DATA IN MAR + 7
GO TO TEST FIFO

FI0000: BRWTFE IMM,4
MODST BR,ADD,SP1,LDMAK

3s:
MEM IMM,0,INCHAR
MEM IMM,0
SP MEMX,SELB,SP14
SP MEMA,SELB,SP15,INCMAR

BRWRTE IMM,7
BRWRTE BR,AANDIR,SP11
BROTAT
MEM BR,SELB,INCHAR
SP BR,SELB,SP2
BRWRTE IMM,MICTRL
MEM IRUS,REFIDA
ALWAYS LUF157

: INIT COUNT
: SAVE TYPE OUT
: FLAG FOR TAB ADDRESS
: IS IN BIT 7
: NEW STATE

MAC RECEPTION ROUTINES

```
1706 ;
1707 ;
1708 ; *****
1709 ; * BEGIN OF FRAME AND END OF FRAME ? *
1710 ; *****
1711 ; READ CHARACTER
1712 ; EXIT
1713 ;
1714 ;
1715 ;
1716 0065b0 FI0100: BRWPE IEUS,PEFIDA
1717 0065b2          ALWAYS LU10
1718 ;
1719 ;
1720 ;
```

KMC RECEPTION ROUTINES

```

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 ; F10200: BRWRITE IMM,10
1736 ; NOUST BP,ADD,SPI,LDMAR
1737 ;
1738 ; MFM IPUS,REFIDA
1739 ; BRWRITE BP,SELA,SP2
1740 ; BRI F10210
1741 ; BRWRITE IMM,INOR
1742 ; ALWAYS LUF157
1743 ; BRWRITE IMM,LFD1
1744 ; ALWAYS LUF157
1745 ;
1746 ;
1747 ;
1748 ;

```

; SAVE DATA IN MAR

; INFO FRAME

; CONTROL FRAME

1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761 006606
1762 006610
1763
1764 006612
1765
1766 006614
1767
1768
1769

```
*****  
: * CONTROL CHARACTER + EOF *  
*****  
: STORE DATA IN MAR + 8  
: SEND MESSAGE UK AND EXIT  
:  
:  
:  
:  
F10300: BRW RTE 1MM,10  
: NODST BR,ADD,SF1,LDWAR  
:  
: MFM IBUS,REFIDA  
: ALWAYS OKMESR  
:  
:  
:  
: SAVE DATA  
:
```

1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782 006616
1783 006620
1784 006622
1785 006624
1786 006626
1787
1788
1789

```
?  
* *****  
* LONG FRAME, DATA  
* *****  
*  
? SAVE DATA IN MAR + 4  
? NEW STATE = LONG FRAME, DATA 2  
? GO TO TEST IF FJFO READY  
?  
?  
?  
?
```

```
FI0400: BRWTFE IMM,4  
MODSI BR,ADD,SPI,LDNAR  
MEM IHUS,REFIDA  
BRWTFE IMM,LFD2  
ALWAYS LUF157  
?  
?  
?
```

```
? SAVE DATA  
? NEW STATE
```

```
1791 ;  
1792 ;  
1793 ; *****  
1794 ; * LONG CONTROL FRAME, DATA 1 OR 2, AND EOF *  
1795 ; *****  
1796 ;  
1797 ; IT IS AN ERROR CASE  
1798 ;  
1799 ; FORCE FCS ERROR IN MAR + 14 ( ERROR STATUS )  
1800 ; BRANCH TO ERROR ROUTINE  
1801 ;  
1802 ;  
1803 ;  
1804 ;  
1805 ;  
1806 ;  
1807 ;  
1808 ;  
  
1803 006630  
1804 006632  
1805 006634  
1806  
1807  
1808  
  
FI0500: BRWRT IMM, FCSERR  
OUT BR, SELB, OBA1  
ALWAYS FI1100  
;  
;  
;
```

1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829

006636
006640
006642
006644
006646

```
*****  
; * LONG CONTROL FRAME, DATA 2 *  
*****  
;SAVE DATA IN MAR + 5  
; NEW STATE = LONG FRAMH , DATA 3  
; GO TO TEST IF FIFO READY  
; ;  
; ;  
; ;
```

```
FI0600: BRWRTI IMM,5  
        MODST  FF,ADD,SPI,LDMAR  
        MEN    IRUS,REFIDA  
        BRWRTI IMM,LFD3  
        ALWAYS LUF157  
; ;  
; ;
```

```
; SAVE DATA 2  
; NEW STATE
```

```
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844 006650  
1845 006652  
1846 006654  
1847 006656  
1848 006660  
1849  
1850  
1851  
  
*****  
* LONG CONTROL FRAME, DATA 3 ; ERROR CASE *  
*****  
; IT IS AN ERROR CASE  
; NEW STATE = ADRPT RX  
; READ FIFO DATA  
; FORCE FCS ERROR IN ERROR STATUS ( MAR + 14)  
;  
;  
FI0800: BRWRTE IMM,ABCK  
SP H,SELB,SP10  
BRWRTE IMM,FCSERR  
OUT BR,SELB,ORA1  
ALWAYS F11000  
;  
;  
;  
; NEW STATE  
; FORCE FCS ERROR
```



```
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863 006062  
1864 006064  
1865  
1866  
1867  
  
; *****  
; * LONG CONTROL FRAME, DATA 3 AND END OF FRAME *  
; *****  
; READ DATA IN SP11  
; SEND MESSAGE ON AND EXIT  
;  
;  
; F10900: SP IHUS,REFIDA,SP11  
; ALWAYS OKMESR  
; ;  
; ;
```

1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882 006666
1883 006670
1884
1885
1886

* ABUFT RECF11JUN *

REAP FIFO DATA
GO TO JEST IF FIFO FREE

F11000: BRWTF IRDS,RE1PA
ALWAYS LRF162

* READ DATA

1898
 1899
 1900
 1901
 1902
 1903
 1904
 1905
 1906
 1907
 1908
 1909
 1910
 1911
 1912

```

; *****
; * ABORT RECEPTION AND END OF FRAME *
; *****
;
; AN ERROR MESSAGE HAS TO BE SEND
;
; READ LAST FIFO DATA
; THE ERROR STATUSES ARE IN NPR6 AND MAR + 15
; CALL GENERAL ERROR ROUTINE
;
;
;

```

```

; READ DATA
; ERROR KMC
; NO TX ERROR
; NEW STATE

F11100: NODST    IRUS,PEF1DA
F11110: BRWTE    IMP,17
        NODST    BR,ADD,SPI,LDMAR
F11120: OUT      MEX,SELB,CRA2
        MFM      IMM,0
        OUT      MEX,SELB,IRA2
        NODST    BR,SELA,SPI,LDMAR
        MFM      IMM,RXFREE
        ALWAYS   ERGEN
;
;
;

```

```

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 AROBT RX
1925 ;
1926 ; SAVE CHARACTER ADDRESS
1927 ;   IN NPK6 ( ADDRESS LOW )
1928 ;   IN NPK7 ( ADDRESS MEDIUM )
1929 ;   IN NISC ( 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 ;
1939 ; 1 - TEST NEW BUFFER FLAG
1940 ; BRWRTI IMM,3
1941 ; MODST BR,ADD,SP1,LDMAR
1942 ;
1943 ; SPBF MEMX,SELB,SP0 ; READ FLAG
1944 ; FI1221 ; RESET STATE
1945 ; MODST BR,SELA,SP1,LDMAR ; NOT YET FLAGGED
1946 ; MEM IMM,INOB
1947 ; ALWAYS LUI0
1948 ;
1949 ;
1950 ;
1951 ;
1952 ; 2 - TEST LAST BUFFER FLAG
1953 ; BRWRTI IMM,3
1954 ; MODST BR,ADD,SP1,LDMAR
1955 ;
1956 ; BRWRTI MEMX,SELB ; READ FLAG
1957 ; BR7 FI1250 ; ERROR
1958 ; BRADDR FI1540 ; PREPARE EXIT ADDRESS
1959 ; SP BR,SELB,SP11 ; IF NON EX MEM
1960 ;
1961 ;
1962 ;
1963 ; 3 - FIRST DMA TO GET CHARACTER ADDRESS
1964 ; CALLSF SP8,TABAD,TABAR2 ; DMA
1965 ; OUT IPUS,INDAT1,OBAA1 ; SAVE DATA
1966 ; BRWRTI IPUS,INDAT2,OBAA2 ; CHECK PARITY
1967 ; BR7 FI1230 ; EVEN ADDRESS
1968 ; BRWRTI IMM,INIF
1969 ; ALWAYS FI1240
1970 ;

```


2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035 007050
2036 007052
2037 007054
2038 007056
2039
2040
2041

```
;  
;  
; *****  
; * ABORT REQUESTED BY PDP AND END-OF-FRAME *  
; *****  
;  
; READ DATA  
; NEW STATE = FREE  
;  
;  
; F11300: MODST IPUS, RFFIDA  
; MODST BR, SELA, SPI, LDMAK  
; MEM . IMM, RXFREE  
; ALWAYS LU10  
; ; ;
```

```
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053 0070b0  
2054 0070b2  
2055 0070b4  
2056  
2057  
2058  
  
*****  
* INFO FRAME, NEW BUFFER, EVEN ADDRESS *  
*****  
; SAVE DATA IN NPRI  
; NEW STATE = NEW BUFFER, SECOND CHARACTER  
;  
;  
;  
FI1400: OUT      IHUS,REFIDA,OIDAT2  
                BRWPIE IMM,INZE  
                ALWAYS LUF160  
;                ;  
;                ;  
;                ;
```

KMC RECEPTION ROUTINES

```

2060 ;
2061 ; *****
2062 ; * INFO FRAME, NEW BUFFER, EVEN ADDRESS, END OF FRAME *
2063 ; *****
2064 ;
2065 ;
2066 ;
2067 ;
2068 ;
2069 ;
2070 ;
2071 ;
2072 ;
2073 ;
2074 ;
2075 ;
2076 ;
2077 ;
2078 ;
2079 ;
2080 ;
2081 ;
2082 ;
2083 ;
2084 ;
2085 ;
2086 ;

F11500: OUT          IRUS,REFIDA,OUTDAI
F11510: BRWRTIE      IMM,221          ; DMA 1 BYTE
F11520: OUT          BR,SELB,ONPR
F11530: BRWRTIE      IRUS,NPR        ; WAIT END OF DMA
      BRO            F11530
      BRWRTIE        IRUS,UBBR
      RPO            F11540          ; NON EX MEM
      ALWAYS         OMESP          ; END OK, MESSAGE, EXIT
;
F11540: BRWRTIE      IMM,17          ; NON EX MEM
      NODST          BR,ADDL,SPI,LDMAR
      MEM            IMM,NOEXME
      BRWRTIE        IMM,100
      SP             IRUS,UBBK,SPU
      OUT            BR,AANDE,OBRR
      ALWAYS         F11255
;
;
;

```


MAC RECEPTION ROUTINES
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

```
*****  
* IMFC 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 = RODY OF BUFFER, EVEN ADDRESS, 1ST CHARACTER  
; BRANCH TO DISPATCHING ROUTINE  
;  
;  
;  
;  
F11600: OUT IRUS,REFIDA,OUTPA2  
BR*RITE IMM,221  
OUT BR,SELB,UNPR  
F11610: BR*WRITE IRUS,NPR  
BR0 F11610  
SP IRUS,IOHA1,SPO  
OUT ER,DECA,UBA1  
BR*RITE IMM,IR1  
ALWAYS LUF160  
;  
;  
;  
;  
; START DMA  
;  
; MAKE ADDRESS EVEN  
;  
; NEW STATE
```

```
2116 ;
2117 ;
2118 ; *****
2119 ; * INFO FRAME, NEW BUFFER, ODD ADDRESS, END OF FRAME *
2120 ; *****
2121 ;
2122 ; LOAD DATA IN NFR3
2123 ; BRANCH TO THE PART THAT INITIATES A DMA FOR 1 BYTE,
2124 ; THEN SENDS A MESSAGE OK
2125 ;
2126 ;
2127 ;
2128 ;
2129 ;
2130 ;
2131 ;

          FI1700: OUI          IGUS,REFIDA,OUTDA2
                ALWAYS FI1510
                ;
                ;
                ;
```

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
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160 007152
2161 007154
2162 007156
2163 007160
2164
2165
2166

;
;
; *****
; * INFO FRAME, NEW BUFFER, 2ND CHARACTER, END OF FRAME *
; *****
;
; LOAD FIFO DATA IN NPR3
; LOAD SAVED DATA IN NPR2
; INITIATE 2 BYTE DMA
; BRANCH TO PIECE OF CODE THAT SENDS A MESSAGE OK
;
;
;
;
;
; F11900: OUT          IHUS,INDAT2,OUTDA1          ; PREPARE DATA
; OUT          IRUS,REFIDA,OUTDA2          ; PREPARE DMA
; BRWRT: IMM,21
; ALWAYS F11520
;
;
;

```

```
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180 007162  
2181 007164  
2182 007166  
2183  
2184  
2185  
  
*****  
* INFO FRAME, BODY OF BUFFER, 1ST CHARACTER *  
*****  
? SAVE DATA IN NPR1  
? NEW STATE = BODY OF BUFFER, 2ND CHARACTER  
? THIS IS THE FIRST MOST COMMON CASE  
  
FI2000: OUT IPUS,REFIDA,OIDAT2  
BRWTE IMM,IR2  
ALWAYS LUF160  
?  
?  
?  
?  
? SAVE DATA  
? NEW STATE
```

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 *  
: *****  
: *AIT FOR LAST DMA  
: * ADD 2 TO CHARACTER ADDRESS  
: BRANCH TO POINT WHERE  
: DATA IS LOADED IN DPR2  
: DMA IS INITIATED FOR 1 BYTE  
: EXIT TO SEND MESSAGE OK  
: *****  
FI2100: CALLSR SPS,RXDMA0,FI1500  
: ; ;
```



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



```

2263 ; *****
2264 ; * INFO FRAME, END OF BUFFER, FIRST CHARACTER *
2265 ; *****
2266 ; *****
2267 ; * WAIT END OF DMA
2268 ; * INITIATE DMA TO SEND SAVED DATA
2269 ; * BRANCH TO ROUTINE THAT GIVES A NEW BUFFER
2270 ; *****
2271 ; *****
2272 ; *****
2273 ; *****
2274 ; *****
2275 SF5,HXDMA0
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287

FI2400: CALLSR SF5,HXDMA0
      OUT
      BRWRTI IMN,221
      OUT
      BR,SELB,ONPR
FI2410: BRWRTI IRUS,IMR
      BRO
      BRWRTI IRUS,UBRR
      BRO
      ALWAYS FI1220
      ;
      ;
      ;

; WAIT DMA
; AND INC ADDRESS
; PREPARE DATA
; REQUEST DMA
; WAIT DMA
; NON EX MEM
```

DMSA DEC/X11 SYSTEM EXERCISER
KMC RELAY ADDRESS IN PAGE 2

MACRO M1113 01-JUL-80 16:42 PAGE 96

2289
2290
2291
2292
2293
2294 007274
2295
2296
2297

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

KNC FIFO OUTPUT READY

2299 ; .SBTTL *KNC* FIFO OUTPUT READY
2300 ; * *****
2301 ; * FIFO OUTPUT READY AND TRANSMISSION *
2302 ; * *****
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 ; LUF00: SP BR, AANDB, SP6 ; UPDATE SCAN
2320 ; LDMAP IMM, 0 ; PAGE 0
2321 ; BRWRTI IMM, 6 ;

2322 ; NODST BR, ADD, SP1, LDMAR ; TYPE OUT
2323 ; SP MEMX, SELB, SP2 ;
2324 ; NODST BR, SELA, SP1, LDMAR ; LOAD STATUS
2325 ; SP MEMX, SELB, SP10, INCMAR ;

2326 ; LUF010: BRADDR LUF020 ; DISPATCH
2327 ; .ALWAY BR, ADD, SP10, P2 ; FREE ?
2328 ; ALWAYS LU20 ; FREE ?
2329 ; ALWAYS LU20 ; BEGIN OF FRAME
2330 ; ALWAYS PRA00 ; BEGIN OF BUFFER ODD
2331 ; ALWAYS LUF030 ; SECOND CHARACTER
2332 ; ALWAYS LUF040 ; FIRST CHARACTER
2333 ; ALWAYS LUF030 ; SHORT FRAME : ADDRESS
2334 ; ALWAYS PDA00 ; SHORT FRAME : CONTROL
2335 ; ALWAYS PDB00 ; LONG FRAME: DATA 3
2336 ; ALWAYS PCD00 ; DATA 2
2337 ; INCMR ; DATA 1
2338 ; ALWAYS PCC00 ; CONTROL
2339 ; ALWAYS PCB00 ; ADDRESS
2340 ; ALWAYS PCA00 ;
2341 ; ALWAYS PFA00 ; TX ABORT
2342 ; ALWAYS PFB00 ; NON EX MEM
2343 ;
2344 ;
2345 ;

LUF030: CALLSR SP5, LOCONT, PBC00 ; LOAD CONTEXT
LUF040: CALLSR SP5, LOCONT, PBD00 ;

2346 ;
2347 ;
2348 ;
2349 ;
2350 ;
2351 ;

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

LUMSU:

BR*RIE INK,MODSTA
SP BK,SELB,SP2
ALWAYS OKMOD

; TYPE OUT

```
2423 *SBTTL *KMC* COMMAND ACKNOWLEDGE
2424 ;
2425 ; *****
2426 ; * COMMAND ACKNOWLEDGE *
2427 ; *****
2428 ;
2429 ; PAGE 2
2430 ; CLEAR COMMAND FLAG
2431 ; SEND MESSAGE OK TO PDP
2432 ;
2433 ;
2434 ;
2435 LUACO: HF*RTI IMM,COMMAN
2436 SP HF,SELH,SP2
2437 ALWAYS UKMESS
2438 ;
2439 ;
2440 ;
```

```
; 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

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

KMC QUEUING MESSAGES

```

2463 .SETL *KMC* QUEUING MESSAGES
2464 ;
2465 ; *****
2466 ; * QUEUING MESSAGES *
2467 ; *****
2468 ; *****
2469 ; *****
2470 ; *****
2471 ; *****
2472 ; *****
2473 ; *****
2474 ; *****
2475 ; *****
2476 ; *****
2477 ; *****
2478 ; *****
2479 ; *****
2480 ; *****
2481 ; *****
2482 ; *****
2483 ; *****
2484 ; *****
2485 ; *****
2486 ; *****
2487 ; *****
2488 ; *****
2489 ; *****
2490 ; *****
2491 ; *****
2492 ; *****
2493 ; *****
2494 ; *****
2495 ; *****
2496 ; *****
2497 ; *****
2498 ; *****
2499 ; *****
2500 ; *****
2501 ; *****
2502 ; *****
2503 ; *****
2504 ; *****
2505 ; *****
2506 ; *****
2507 ; *****
2508 ; *****
2509 ; *****

ERRMC: OUT BR,SELB,IOA2 ; KMC ERROR MESSAGE
CALLSR SP5,OUTME1,ERKMC1 ; NOT COMPLETE

ERKMC: OUT BR,SELB,IOA2 ; KMC ERROR MESSAGE
ERKMCB: CALLSR SP5,MESCOM,ERKMC1 ; COMPLETE, NO CLEAR TAB

NOBAD: BRWRTE IMM,200 ; NO BAD ADDRESS
CALLSR SP5,OUTME1,ERKMC2 ;

OKMES: MODST BR,SELA,SP1,LDWAR ; TX MESSAGE OK
MFM IMM,KFREE ;
OKMESS: CALLSR SP5,MESCOM,MESCSR ;

OKMESR: MODST BR,SELA,SP1,LDWAR ; RX MESSAGE OK
MFM IMM,EXFREE ;
BRADDR MESCSR ;
SP BR,SELB,SP5 ;

OKMESQ: BRWRTE BR,SELA,SP2 ; CLEAR TAB ADDRESS
BR7 MESCO1 ;
ALWAYS MESCOM ;

OKMOD: CALLSR SP5,OUTME1,OKMS ; MODEM STATUS

ERGEN: BRADDR ERGE00 ; GENERAL ERROR MESS
SP BR,SELB,SP5 ; PREPARE RETURN ADDRESS
BRWRTE I0US,IOA2 ; GET ERROR 2 BYTE
RRO MESCOM ; NO BUFFER, NO DELETE
ALWAYS UKMESQ ; DELETE BUFFER IF REQ
; (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 MARK ON OVERALL COUNT
; LOAD OVERALL COUNT MINUS TEMPORARY COUNT IN SP14, SP15
; LOAD 1ST AND 2ND CHARACTERS IN SP9, SP10
; QUEUE MESSAGE
;
;

```

```

MESCOM: BRWRTE IMM,3          ; CLEAR TAB ADDRESS
        NODST BR,ADD,SP1,LDMAR
        MEM IMM,0
        BRWRTE IMM,4
        NODST BR,ADD,SP1,LDMAR
        BRWRTE BR,SELA,SP14
        SP MEMX,SELB,SP14
        SP RF,SUB,SP14,INCMAR
        BRWRTE BF,SELA,SP15
        SP MEMX,SELB,SP15,INCMAR
        SP BR,SUBC,SP15,INCMAR
        ;
        SP MEMX,SELB,SP9,INCMAR
        SP MEMX,SELB,SP10

```

```

; LOW COUNT
; HIGH COUNT
; 1ST CHARACTER
; 2ND CHARACTER

```

2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560 007620
2561 007622
2562 007624
2563 007626
2564 007630
2565 007632
2566 007634
2567 007636
2568 007640
2569 007642
2570 007644
2571
2572 007646

* LOAD CSR MESSAGE IN QUEUE *

POINT MAR ON MESSAGE QUEUE, PAGE 2

CASE 1 : LONG CONTROL FRAME

DATA 1 IN SP14

DATA 2 IN SP15

DATA 3 IN SP11

CASE 2 : INFORMATION FRAME

BYTE COUNT IN SP14 AND SP15

BIT COUNT IN SP11

LOAD SP14, 15, 11 IN MAR

OUTME1: LDHAP IMM,2 ; LAST MESSAGE
LDMA IMM,2776 ; POINTER
SP MEMX,SELB,SP0 ;
SP IMM,10,SP8 ;
MEM MEMX,ADD,SP8 ; INC POINTER
NODST BR,SELA,SP0,LDMAR ; POINT ON MESSAGE
LDHAP IMM,3 ;
MEM MEMX,SELA,SP14,INCMAR ;
MEM MEMX,SELA,SP15,INCMAR ;
MEM MEMX,SELA,SP11,INCMAR ; CASE 1: DATA 3
MEM MEMX,SELA,SP11,INCMAR ; CASE 2: BIT COUNT
RTNSUB SP5,P3 ;

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

2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603 007602
2604
2605 007664
2606 007606
2607 007670
2608 007672
2609 007674
2610
2611 007676

* LOAD KNC ERROR MESSAGE IN QUEUE *

? LOAD FROM FLAG IN SP4 WITH LINE NUMBER
? ERROR IS IN SP3
? LOAD OK ERROR IN MEMORY
? THEN GO TO QUEUE CSR MESSAGE

ERKMC1: BRWPT: IMM,300
ERKMC2: SP BR,AOR,SP4
MEM IMM,0,INCMAR
MEM IRUS,IOBA2,INCMAR
MEM IMM,0,INCMAR
MEM IMM,0
? ALWAYS MESSR

```
2613 ;
2614 ;
2615 ;
2616 ;
2617 ;
2618 ;
2619 ;
2620 ;
2621 ;
2622 ;
2623 ;
2624 ;
2625 ;
2626 ;
2627 007700 ;
2628 007702 ;
2629 007704 ;
2630 ;
2631 ;
2632 ;
2633 ;
2634 ;
2635 ;
2636 007706 ;
2637 ;
2638 007710 ;
2639 007712 ;
2640 ;
2641 ;
2642 007714 ;
2643 ;
2644 007716 ;
2645 007720 ;
2646 007722 ;
2647 007724 ;
2648 007726 ;
2649 ;
2650 ;
2651 ;
2652 ;
2653 007730 ;
2654 ;
2655 007732 ;
2656 007734 ;
2657 ;
2658 ;
2659 007736 ;
2660 ;
2661 ;
2662 ;
2663 ;
2664 ;
2665 ;
2666 007736 ;
2667 ;
2668 ;

*****
* LOAD ERROR MESSAGE IN QUEUE WHEN DETECTED BY L.U. *
*****

; SP2: TYPE OUI
; SP4: LINE NUMBER
; SP13: STATUS HIGH
; MPR5: ERROR COMING FROM STATUS LOW (ERR.1)
; MPR6: ERROR COMING FROM RX QUEUE (ERR.3)
; MPR7: ERROR DETECTED BY KMC (ERR.2)

ERGE00: MEM IRUS,IJRA2 ; LOAD ERROR 1
BRWTFE IMM,300 ;**** 26-OCT-79 ; ERROR FLAG ONLY
SP BR,AORP,SP4 ;
;
; SP IMM,210,SP8 ;
CALLSB SP5,ERMAS,MC41 ; INIT ERROR CLASS 4
CALLSB SP5,ERMAS,MC31 ; ERROR CLASS 4
CALLSB SP5,ERMAS,MC21 ; CLASS 3
CALLSB SP5,ERMAS,MC11 ; CLASS 2
INCKA ; CLASS 1
;
MEM IRUS,IORA2 ; LOAD ERROR 2
BRWTFE IMM,300 ;
SP BR,SELB,SP8 ;
CALLSB SP5,ERMAS,MC12 ; INIT ERROR CLASS 1
INCKA ; ERROR CLASS 1
;
MEM IMM,300 ;
SP IBUS,RELUAD,SP3 ;
MEM MEMX,AANDB,SP3 ;
SP IRUS,IORA1,SP3 ;
MEM MEMX,AORB,SP3 ;
SP IMM,210,SP8 ;
CALLSB SP5,ERMAS,MC43 ; LOAD ERROR 1
CALLSB SP5,ERMAS,MC33 ; INIT ERROR CLASS 4
CALLSB SP5,ERMAS,MC23 ; ERROR CLASS 4
INCKA ; ERROR CLASS 3
;
MEM IMM,0 ;
ALWAYS MESCSR ;
;
;
SP BR,SELB,SP3 ; LOAD MASK
NDDST MEMX,AORNB,SP3 ; COMPARE
Z ERMAS1 ;
BRWTFE IMM,200 ;
BRWTFE BR,AORP,SP8 ;
SP BR,AORP,SP4 ;
;
ERMAS1: SP BR,TWOA,SP8 ; LOAD CLASS IN SP4
; RTNSUF SP5,P3 ; INCREMENT CLASS
; ****
```

2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683 007736
2684 007740
2685
2686
2687 007742
2688 007744
2689 007746
2690 007750
2691 007752
2692 007754
2693 007756
2694 007760
2695 007762
2696
2697 007764
2698 007766
2699 007770
2700
2701 007772

* LOAD MESSAGE IN QUEUE *

; THIS FORTIME UPDATES THE QUEUE
; AND LOADS MESSAGE IN MEMORY
; -TYPE OUT WITHOUT RDYO IS IN SP2
; -LINE NUMBER IS IN SP4
; THEN BRANCH TO ADDRESS LOADED IN SP5

MESCSR: LDMAP IMM,2
MODSI RR,INCA,SPU,LDMAR

MEM MEMX,SELA,SP10,INCMAR
MEM MEMX,SELA,SP9,INCMAR
MEM MEMX,SELA,SP4,INCMAR
BRWRTI IMM,7
SP RR,AAADB,SP4
MEM IMM,57
SP MEMX,AAADR,SP2
MEM IMM,RDYO
MEM MEMX,ADRB,SP2

; CLEAR FLAG ERROR IF ANY

BRWRTI IMM,100
SP TRUS,UBBR,SP0
OUI RR,AAADR,ORR
; ALWAYS LUI0
; RESET NON EX MEMORY IF ANY

2703
2704
2705
2706
2707
2708
2709
2710

.SBTTL *KMC* SUBROUTINES

;
;
;
;
;
;
;

* SUBROUTINES SECTIONS *

```

2712 * KMC+ SUBROUTINE
2713
2714
2715 *****
2716 * TABLE ADDRESS *
2717 *****
2718
2719 ; THIS ROUTINE
2720
2721 ; -LOAD MAR WITH ADDRESS OF TABLE ADDRESS
2722 ; -PREPARE DMA BY LOADING TABLE ADDRESS IN
2723 ; RPA REGISTERS
2724 ; -ALSO INCREMENTS TABLE ADDRESS BY 2 FOR
2725 ; NEXT STEP
2726 ; -REQUESTS DMA IN
2727 ; -WAITS FOR END OF DMA
2728 ; -CHECKS IF MEMORY INEXISTENT; IF IT IS THE
2729 ; CASE, EXIT TO ADDRESS IN SP11
2730
2731 ; RETURN ADDRESS IS IN SP8
2732 ; RETURN ADDRESS IF NON EX-MEMORY IS IN SP11
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759

TABAD:  NOUJST  BR,INCA,SP1,LDMAR           ; LOAD MAR
        SP     MEMX,SELB,SP14           ; ADDRESS LOW
        OUT   MEMX,SELB,IBAI
        BRWRT  IMM,2
        MEM   BR,ADD,SP14,INCMAR       ; ADD 2
        SP     MEMX,SELB,SP15         ; ADDRESS MEDIA
        OUT   MEMX,SELB,IBAI
        MEM   BR,APLUSC,SP15,INCMAR
        OUT   MEMX,SELB,ONPR
        C     TABAD2
TABAD1:  BRWRT  IEUS,NPR
        BRO   TABAD1
        BRWRT  IEUS,UBAR
        BRO   SELA,SP11,P1
        ;
        RTNSUR  SP8,P3
TABAD2:  BRWRT  IMM,4
        SP     MEMX,SELB,SP14
        MEM   BR,ADD,SP14
        ALWAYS  TABAD1
TABAR1:  ALWAYS  PRAIS
TABAR2:  ALWAYS  F11225
;
;
;

```


2761
2762
2763
2764
2765
2766
2767
2768 010046
2769 010050
2770 010052
2771 010054
2772 010056
2773 010060
2774 010062
2775 010064
2776 010066
2777
2778 010070
2779 010072
2780 010074
2781 010076
2782
2783 010100
2784 010102
2785 010104
2786 010106
2787
2788
2789
2790

```
;  
; *****  
; * WAIT DMA AND INCREMENT NPR ADDRESS BY 2 *  
; *****  
;  
; RXDMA0: BFWTE IBUS,NPR ; WAIT DMA  
; BPU RXDMA0  
; BFWTE IRUS,UBHK  
; BPO F11540  
; RXDMA1: SP IRUS,IORAI,SPO ; NON EX MEM  
; SP BR,INCA,SPO  
; OUT BR,INCA,ORA1 ; INC ADDRESS LOW  
; C RXDMA2  
; RTNSUB SP5,P2  
;  
; RXDMA2: SP IRUS,IOKA2,SPO ; INC ADDRESS MEDIUM  
; OUT BR,INCA,ORA2  
; C RXDMA3  
; RTNSUB SP5,P2  
;  
; RXDMA3: BFWTE IMP,4 ; INC ADDRESS HIGH  
; SP IRUS,UBHK,SPO  
; OUT BR,ADD,ORR  
; RTNSUB SP5,P2  
;  
; ;  
; ;  
; ;  
; ;
```

```

*KKC* TRANSMISSION SUBROUTINES
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825

*KKC* TRANSMISSION SUBROUTINES
*****
* TRANSMISSION SUBROUTINES *
*****

*****
* INITIALISATION *
*****

THIS ROUTINE CHECKS IF
L.U. PRESENT
L.U. NOT BUSY
IF NOT OK, IT OUTPUTS AN ERROR MESSAGE
TO PUF AND FX11
ELSE, IT RETURNS TO SPS

TXINI: SPBK IMM,0,SPG,LDMAPG ; PAGE 0
SP NEY,SELB,SP10 ; TX STATUS
COMP FE,SP10 ; BUSY
C TXINI2 ; READ LINE UNIT ADDRESS
TXINI3: BRWTE IRUS,RELUAD ; ON LINE
.BR4 SELA,SP5,P1 ; NOT PRESENT OR OFF LINE
TXINI1: FEWTE IMM,NOTPRE
ALWAYS ERKMC

TXINI2: BRWTE IMM,LUBUSY
ALWAYS FEKPC
;
;
;

```

```

* KMC *
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846 010134
2847 010136
2848
2849 010140
2850 010142
2851 010144
2852 010146
2853 010150
2854 010152
2855
2856 010154
2857 010156
2858
2859 010160
2860 010162
2861 010164
2862 010166
2863
2864 010170
2865 010172
2866 010174
2867
2868 010176
2869 010200
2870 010202
2871 010204
2872 010206
2873 010210
2874 010212
2875
2876 010214
2877
2878 010216
2879 010220
2880 010222
2881
2882
2883

*****
* READ PASE ADDRESS *
*****

; SPO CONTAINS PAGE
; THIS ROUTINE IS CALLED AT INITIALISATION
; OF TX OF INFO FRAME
; OF TX OF LONG CONTROL FRAME
; OF RX OF INFO FRAME

; IT CHECKS IF PAD EXISTS, IF NOT EXIT IT DOES
; FIRST CVA AND LOAD 2 BYTES
; IN PAR (PAR IS SUPPOSED TO BE READY)
; IT DOES SECOND DMA AND LET THE BYTES
; IN NFF

; IT RETURNS TO ADDRESS IN SP5

; FIND BAD

; NO BAD
; RO DMA

; WAIT DMA
; STORE DATA

; INC ADD LOW

; INC ADD MED

; RO DMA

; INC ADD HIGH

```

```

BADIN: LDWAP IMM,2
LDMA IMM,BADA
;
; OUT MEMX,SELE,IBAI,INCMAR
; OUT MEMX,SELE,IBAI,INCMAR
BRWRTI MEMX,SELE,IBAI,INCMAR
BRO BADI3
ALWAYS NOBAD
BADI3: OUT DR,SELE,ONPR
;
; MODST MEMX,SELA,SPO,LDWAPG
; MODST PR,INCA,SPI,LDNAR
;
; BRWRTI IEUS,NPF
; BRO BADI0
; MEM IEUS,INDAT1,INCMAR
; MEM IEUS,INDAT2,INCMAR
;
; SP IEUS,IIBAI,SPO
; SP BR,INCA,SPO
; OUT BR,INCA,IBAI
;
; SP IEUS,IIBAI,SPO
; OUT BR,APLUSC,IBAI
; SP IEUS,NPR,SPO
; C BADI2
; OUT BR,INCA,ONPR
BADI1: BRWRTI IEUS,NPF
; BRO BADI1
;
; RTNSUB SP5,P1
;
; BRWRTI IMM,5
; OUT BR,ADD,ONPR
ALWAYS BADI1
;
;

```

```

2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936

*****
* GET CHARACTER COUNT *
*****

;
;
; THIS ROUTINE
; CALLS ROUTINE TO PERFORM DMA
; SAVES COUNT LOW AND HIGH IN SP14 AND 15
; CHECKS LAST BUFFER FLAG
; IF SET (=0), SHOWS HIGH COUNT IN
; SP15 BY BITS AND SAVES BIT COUNT
; + LAST BUFFER FLAG IN SP3
; PREFACE IFR COMMAND WITH ADDRESS HIGH
; AND SAVE IN BRG
;
; RETURN TO ADDRESS IN SP5
;
; THIS ROUTINE IS CALLED WHEN A NEW BUFFER IS
; TO BE STARTED FOR TX OR RX
;
;
DMAC: CALLSB SP8,IABAD ; DMA
SP IFR,INDAT1,SP14,INCMAR ; LOW COUNT
MEM MEMX,ADD,SP14,INCMAR ; OVERALL
; COUNT
SPBR IFR,INDAT2,SP15 ; HIGH COUNT
BFOTAI
BR4 DMAC10
;
;
SP BR,SELB,SP3 ; LAST BUFFER
BWRITE IMM,3 ; HIGH COUNT
SP BR,AAAND6,SP15 ; OVERALL
MEM MEMX,ADDC,SP15,INCMAR ; COUNT
;
BWRITE IMM,16 ; BIT COUNT
SP BR,AAANDH,SP3,INCMAR ; AND
BWRITE IMM,EOF ; END OF
SF BR,AOKB,SP3 ; FRAME
ALWAYS HIPAAD
;
;
DMAC10: BWRITE IMM,27 ; NOT LAST BUFFER
SP BR,AAAND6,SP15 ; HIGH COU
MEM MEMX,ADDC,SP15,INCMAR ; OVERALL
BWRITE IMM,0,INCMAR ; COUNT
SP BR,SELB,SP3 ; ALL CASES HIGH PART
; OF ADDRESS
;
CONTINUE NEXT PAGE

```

*RAC+ TRANSMISSION SUBROUTINES

2938
2939
2940
2941
2942
2943
2944
2945 010300
2946 010302
2947 010304
2948 010306
2949 010310
2950 010312
2951 010314
2952 010316
2953 010320
2954
2955

HIPAAD: SP
HIPA10: BRWTE
BRWTE
BROTAT
BROTAT
BROTAT
BROTAT
SP
RTNSUH
;
;

IRUS,INDAT2,SP8
IRM,300
BR,AANDR,SP8

BK,SELB,SP8
SP5,PI

* HANDLING OF HIGH PART OF ADDRESS *

```

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
TRANSMITTING DATA, AFTER AN
INTERRUPTION
THE CONTEXT IS : SAVED DATA (2 BYT)
COUNT (2 BYT)
BUFFER ADDRESS (3 BYT)
INIT BLOCK COUNT
THE MAP POINTS ON BLOCK ADD + 7 AFTER
THE RETURN ADDRESS IS IN SP5

LOCONT: BPWRTF IMX,10 ; INIT
NDUST 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 ; ADDRESS
OUT MEMX,SELB,IBAI,INCMAR
OUT MEMX,SELB,IBA2,INCMAR
OUT MEMX,SELB,OUTDA1
;
BPWRTF IMR,FLOCK ; INIT BLOCK COUNT
SP BR,SELB,SP9 ;
;
BPWRTF IMX,7 ; RELOADMAR
NDUST 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 010374
3016 010376
3017 010378
3018 010400
3019 010402
3020
3021 010404
3022 010406
3023
3024 010410
3027

*****
* END OF FRAME DETECTED BY LINE 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

; RECEPTION
; INFO, NO BUFFER
; EXIT

; TEST IF BUFFER
; BUFFER : EXIT
; ERROR
; NO BUFFER AVAILABLE
; TYPE OUT IN SP2

FS00:
LDMAP IMM,1
SP MEMX,SELB,SP10,INCMAR
BRWRTI IMM,INOB,INCMAR
MODST HF,SUBOC,SP10,INCMAR
Z FS10
ALWAYS SC30
;

FS10:
BRWRTI MEMX,SELB
BR0 SC30
BRWRTI IMM,17
MODST HF,ALL,SP1,LDMAR
MEM IMM,NORUPF
;
BRWRTI IMM,RXINPO
SP HF,SELR,SP2
;
ALWAYS FI1255

```

 * TRANSMISSION SUBROUTINES *

```

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

```

```

*SUBTIL *KRC* BRKPOINTS DEFINITION
NPMPFS = 4
.SAV=
.EASJAK1+<<1024.-<(NRKKS*2)>>*2>
SBRKNS=0
.PEPT NRKKS
.IRF SSNDRA,<\SNBRKS>
OVI IMM,SSNDRF*20+1,OMAIN
ALWAYS
SBRKKS=SNBRKS+1
.ENDM

```

```

*BRKPTS DEFINITION
KMCODE: KMCODE
.END

```

```

*BRKPTS DEFINITION
KMCODE: KMCODE
.END

```

```

000004
010412
010514
000000
000004

```

```

*BRKPTS DEFINITION
NPMPFS = 4
.SAV=
.EASJAK1+<<1024.-<(NRKKS*2)>>*2>
SBRKNS=0
.PEPT NRKKS
.IRF SSNDRA,<\SNBRKS>
OVI IMM,SSNDRF*20+1,OMAIN
ALWAYS
SBRKKS=SNBRKS+1
.ENDM

```

```

*BRKPTS DEFINITION
KMCODE: KMCODE
.END

```

```

*BRKPTS DEFINITION
KMCODE: KMCODE
.END

```

```

*BRKPTS DEFINITION
KMCODE: KMCODE
.END

```

000004
010412
010514
000000
000004

DMSA DECA11 SYSTEM FACETS
SYMBOL TABLE

AMIB = 00020	PAR2 = 003740R	DEBUT1 = 004536R	FI1500 = 007066R
ARUF = 00012	BEUUN = 000003	DECA = 060160	FI1510 = 007070R
ARCDEF = 00013	HGFRA = 000002	DISCO = 000003	FI1520 = 007072R
ARORF = 00014	RFGIN = 000000R	DISF = 000234	FI1530 = 007074R
ABURRF = 00015	BFF = 000001	DVAC = 010224R	FI1540 = 007106R
ABUR1 = 00100	BI0 = 000001	DVACR = 006144R	FI1600 = 007124R
ARUF1 = 003770R	BI11 = 000002	DVAC10 = 010266R	FI1610 = 007132R
ABUF0 = 003776R	BI10 = 002000	DW100 = 000306R	FI1700 = 007146R
ACSR = 00052R	BI111 = 004000	DW101 = 000624R	FI1900 = 007152R
ACSREX = 004004R	BI12 = 010000	DW102 = 001222R	FI2000 = 007162R
ADJ = 000000	BI113 = 020000	DV111 = 000014R	FI2100 = 007170R
ADDC = 000020	BI114 = 040000	ECSF = 000000	FI2200 = 007176R
ADDUCN = 004012R	BI15 = 100000	ENDFSS = 104402	FI2210 = 007176R
ADDR = 000005R	BIT2 = 000004	ENDS = 104403	FI2220 = 007232R
ALCIRD = 000400	BIT3 = 000010	EOF = 000240	FI2300 = 007240R
AMASK = 000022	BIT4 = 000020	ERGEN = 007554R	FI2400 = 007246R
AMES00 = 003274R	BIT5 = 000040	ERGE00 = 007700R	FI2410 = 007262R
AMES01 = 003302R	BIT6 = 000100	ERKMC = 007476R	FREE = 000104R
AMES02 = 003310R	BIT7 = 000200	ERKMA = 007500R	FS00 = 010356R
AMES03 = 003316R	BIT8 = 000400	ERKMC = 007500R	FS10 = 010372R
AMES04 = 003322R	BIJ9 = 001000	ERKMC = 007500R	GFAPAS = 104413
AMES05 = 003330R	BLOCK = 000040	ERKMC2 = 007664R	GOOD = 003732R
AMES06 = 003336R	BOF = 000300	ERKMC2 = 007664R	GMOC = 000015
AMES07 = 003344R	BR = 060000	ERKMC2 = 007664R	GRU = 003174R
AMES08 = 003350R	BREKAS = 104407	ERKMC2 = 007664R	GMBUFS = 104412
AMES09 = 003356R	BPUCON = 002000	ERRMS1 = 007736R	HIPAAD = 010300R
AMES10 = 003364R	BR1 = 000012R	ERRMS = 104410	HIPAD = 010302R
AMES11 = 003372R	BR1CON = 002400	ERRMS = 104410	IBA1 = 002004
AMES12 = 003400R	BR2 = 000013R	EXITS = 104400	IBA2 = 002005
AMES13 = 003404R	BR4CON = 003000	FCSEK = 000040	IBUS = 020000
AMES14 = 003412R	BR7CON = 003400	FICAR = 000005	IB1 = 000026
AMES15 = 003420R	BUFIN = 004134R	FICTR = 000020	IB1EF = 000027
AMES16 = 003424R	BUFKMC = 001144R	FI0000 = 006524R	IB1EF = 000026
ADRB = 000300	BUFOVE = 000002	FI0100 = 006560R	IR2 = 000030
AORNB = 000240	BYCO = 004020R	FI0200 = 006564R	IB2EF = 000031
APLUSC = 060100	CALLPC = 004026R	FI0210 = 006602R	IDLF = 000000
ASB = 000056R	CCOND = 001000	FI0300 = 006606R	IE1 = 000032
ASEL0 = 004050R	CDDATAS = 104414	FI0400 = 006616R	IE1EF = 000033
ASEL10 = 004060R	CLEKMC = 002420R	FI0500 = 006630R	IE2 = 000034
ASEL12 = 004062R	CLK = 004014R	FI0600 = 006636R	IE2EF = 000035
ASEL14 = 004064R	CLK1 = 004016R	FI0800 = 006650R	IB1A1 = 000100
ASEL16 = 004066R	COMP = 000007	FI0900 = 006662R	IB1A2 = 000120
ASEL2 = 004052R	COMMAN = 000007	FI1000 = 006666R	IMM = 000000
ASEL4 = 004054R	CSRA = 000050R	FI1100 = 006672R	IMDX = 000000
ASEL6 = 004056R	CSREX = 004104R	FI1110 = 006674R	INCA = 060060
ASIA1 = 000054R	CTRLEF = 000003	FI1120 = 006700R	INCRAR = 014000
ASIA2 = 000060R	DATA12 = 003746R	FI1200 = 005714R	INCON = 100000
AXURB = 000320	DATA3 = 003750R	FI1220 = 006732R	INDATI = 000000
BAD = 003734R	DATCKS = 104417	FI1221 = 006740R	INDATI2 = 000020
BADA = 000265	DATERS = 104405	FI1225 = 006754R	INFO1 = 003752R
BADIN = 010134R	DATI = 000001	FI1230 = 006770R	INFO2 = 003762R
BADIN0 = 010160R	DATI1 = 000003	FI1240 = 006772R	INIT = 000022R
BADIN1 = 010210R	DATI2 = 000021	FI1245 = 007002R	INOH = 000016
BADIN2 = 010216R	DATOB = 000223	FI1250 = 007022R	INQEF = 000017
BADIN3 = 010152R	DATOBH = 000223	FI1255 = 007030R	INSTH1 = 003731R
BANOLU = 000020	DATOH = 000023	FI1300 = 007040R	INSTLO = 003730R
BAR1 = 003736R	DEBUT = 004534R	FI1400 = 007060R	INTPLG = 003724R
			INTRDI = 002674R
			LUBUSY = 000200
			LUER0 = 007372R
			LUER10 = 007410R
			LUER30 = 007420R
			LUER40 = 007426R
			LUF10 = 005206R
			LUF120 = 005250R
			LUF130 = 005254R
			LUF140 = 005260R
			LUF150 = 005344R
			LUF151 = 005346R
			LUF152 = 005356R
			LUF153 = 005374R
			LDWAPG = 004000
			LDMAR = 010000
			LFADDR = 000014
			LFCIRL = 000013
			LFDAT1 = 000012
			LFDAT2 = 000011
			LFDAT3 = 000010
			LFD1 = 000004
			LFD1EF = 000005
			LFD2 = 000006
			LFD2EF = 000007
			LFD3 = 000010
			LFD3EF = 000011
			LIN = 003742R
			LINAUT = 002100R
			LINENM = 100060
			LINUMB = 003744R
			LOCONT = 010322R
			LOFICO = 002014
			LOFIDA = 002015
			LOLUAD = 002010
			LOOPCN = 004024R
			LOSTHI = 002017
			LOSTLO = 002016
			LUACO = 007446R
			LUBUSY = 000200
			LUER0 = 007372R
			LUER10 = 007410R
			LUER30 = 007420R
			LUER40 = 007426R
			LUF10 = 005206R
			LUF120 = 005250R
			LUF130 = 005254R
			LUF140 = 005260R
			LUF150 = 005344R
			LUF151 = 005346R
			LUF152 = 005356R
			LUF153 = 005374R

SYMBOL TABLE

LUF157	005404K	MIDNAM	000000K	PEA20	005712K	PSW	177776	SC417	005136R
LUF160	005410K	MODSP	000206K	VPA30	005716R	PUSH	005746	SC418	005142R
LUF161	005412K	MODSTA	000003	FR000	005740R	PUSH2	024646	SC42	005150R
LUF162	005416K	MODT	000003	PF010	005747R	P0	000000	SC43	005160R
LUF163	005416K	MODV	000000	PFC20	005750R	P00	004716K	SC430	005174R
LUF164	005424K	MDCMS	104411	PR000	005772K	P1	004900	SC44	005200R
LUF165	005454K	MSGS	104416	PA010	006002K	F10	004754R	SC50	005016R
LUF168	005466K	MSG	104406	PRD20	006010R	P2	010000	SECHAR	000004
LUF169	005474K	MSJLN	000357	PRL30	006014K	P20	005560R	SELA	060200
LUF00	007276K	MSIOU	000367	PF040	006032F	F3	014000	SELB	000220
LUF010	007314K	MNRKS	000004	PR050	006042F	UUES	104401	SFADDR	000006
LUF020	007320K	NRBAD	007506K	PR060	006062F	KARNDM	003040K	SFCTRL	000007
LUF030	007356K	NRBUFF	000001	PRD70	006072F	KARHAD	003034K	SHFTBR	001400
LUF040	007364K	NREXME	000004	PBD80	006102F	KARJUM	003036K	SMAS	000001
LUF050	007454K	NRX	000010	PRD90	006110K	RGFUEA	000070R	SPBRX	003400
LUMSU	007440F	NULPRE	000010	PR095	006116K	RRUEFA	000066K	SPDINT	000024R
LUU	004576K	NPR	100200	FRD98	006126K	KRUF52	000072K	SPSIZ	000040
LU10	004576K	ORA1	002006	PR000	005564R	KRUFVA	000064R	SPX	003000
LU20	004604K	ORA2	002007	PR000	005612R	R0Y1	000200	SPO	000000
LU30	004514K	OPR	001011	PCA00	006170R	R0Y0	000020	SP1	000001
LU40	004636K	OCOM	100040	PCB00	006220R	REF1CO	000300	SP10	000012
MAIN	100020	OJDAT1	002000	PCC00	006222K	REF1DA	000320	SP11	000013
MAP22	104315	OJDAT2	002001	PCD00	006230R	RELUAD	000200	SP12	000014
MASK	003726K	OINCOM	001000	PC00	006146R	RESCAN	000240	SP13	000015
MC11	000347	OKMES	007516K	PDA00	006266K	RESHI	000360	SP14	000016
MC12	000020	OKMESG	007540R	PDF00	006250R	KESTI0	000340	SP15	000017
MC13	000377	OKMESR	007530R	PF00	006370R	KESTY	000250K	SP16	000016
MC21	000337	OKMESS	007522R	PF10	006404K	RQCSR	002600R	SP17	000017
MC23	000375	OKMOD	007546K	PE10	006417R	RQCSR1	002626K	SP2	000002
MC31	000376	OKMS	007650K	PE20	006442R	RQCSK3	002672R	SP3	000003
MC33	000377	OLINEM	001003	PFA00	006436R	RSTRT	000062R	SP4	000004
MC41	000075	OMAIN	001010	PFB00	006444K	RXDI	000007	SP5	000005
MC43	000075	ONPR	001010	PF010	006442R	RXDWA0	010046R	SP6	000006
MEM1	000075	OPCON	001002	PG00	006424R	RXDWA1	010056R	SP7	000007
MEMX	040000	OPEN	000000	PG10	006500R	RXDWA2	010070R	SP8	000010
MESCU0	007574R	OPORT1	001004	PH00	006512R	RXDWA3	010100R	SP9	000011
MESCU1	007566K	OPORT2	001005	PHU1	006520R	RXFREE	000000	SRI	000016R
MESCSH	007336K	OPORT3	001006	PJ00	006522R	RXIN	000000	START	000206R
MESE	003064K	OPORT4	001007	PJRCS	000004	RXINFO	000000	STAT	000020R
MESS00	003430K	OUTDA1	002002	PJ00	005524K	RXLOC0	000002	STSCAN	002012
MESS01	003437K	OUTDA2	002003	FK00	005504R	RXON	000004	SUB	000340
MESS02	003446K	OUTME1	007620K	FKU1	005512R	RXSHCO	000001	SUBC	000040
MESS03	003455K	OXREG0	002010	POPSP	005726	R6	0000006	SUBDC	000360
MESS04	003501R	OXREG1	002011	POPSP2	022626	F7	0000007	SVR0	000032R
MESS05	003515K	OXREG2	002012	FOR11	100160	SHADR	000052R	SVR1	000034R
MESS06	003531K	OXREG3	002013	FOR12	100120	SCREL	007274K	SVR2	000036R
MESS07	003550R	OXREG4	002014	FOR13	100140	SC00	004664R	SVR3	000040R
MESS08	003566K	OXREG5	002015	FOR14	100160	SC28	004656K	SVR4	000042R
MESS09	003603K	OXREG6	002016	PPTY0	000000	SC30	004702R	SVR5	000044R
MESS10	003630K	OXREG7	002017	PRY1	000040	SC31	004706R	SVR6	000046R
MESS11	003640K	PASCNT	000026R	PRY2	000160	SC40	005022R	TABAD	007774R
MESS12	003650R	PASS	002074K	PRY3	000140	SC409	005060R	TABAD1	010020R
MESS13	003664K	PA00	006322R	PRY4	000200	SC410	005042R	TABAD2	010032R
MESS14	003703K	PA001	006354K	PRY5	000240	SC411	005064R	TABAD3	010044R
MF10	000217	PR00	000000	PRY6	000280	SC412	005086K	TABAD4	010056R
MF10A	000337	PR001	000000	PRY7	000320	SC413	005108K	TABAD5	010068R
MF10B	000337	PR002	000000	PRY8	000360	SC414	005130K	TABAD6	010080R
MF10C	000337	PR003	000000	PRY9	000400	SC415	005152K	TABAD7	010092R
MF10D	000337	PR004	000000	PRY0	000440	SC416	005174K	TABAD8	010104R
MF10E	000337	PR005	000000	PRY1	000480				
MF10F	000337	PR006	000000	PRY2	000520				
MF10G	000337	PR007	000000	PRY3	000560				
MF10H	000337	PR008	000000	PRY4	000600				
MF10I	000337	PR009	000000	PRY5	000640				
MF10J	000337	PR010	000000	PRY6	000680				

SYMBOL TABLE
SYMBOL = 000000
LRAA = 000020
LWCA = 000120
LWAC = 000140
LXAPU = 000115
LXDI = 000106
LXLJ = 000104
LXINFC = 000004
LXLJJ = 010124
LXLJ2 = 010150
LXLJ3 = 010120
LXLQ = 000004
LXLUQU = 000006
LXLM = 000015
LXDP = 000005
LXSH = 000005
LXSHC = 000005

MACRO M1113 01-JUL-80 PAGE 114-3

WTIN = 000002
WTOU = 000001
XFLAG = 00005R
XMASK = 000021
XREG0 = 000200
XREG1 = 000220
XREG2 = 000240
XREG3 = 000260
XREG4 = 000300
XREG5 = 000340
XREG6 = 000360
ZCOND = 001400
\$NBK = 000004
\$.SAV = 010412R

WTIN = 000002
WTOU = 000001
XFLAG = 00005R
XMASK = 000021
XREG0 = 000200
XREG1 = 000220
XREG2 = 000240
XREG3 = 000260
XREG4 = 000300
XREG5 = 000340
XREG6 = 000360
ZCOND = 001400
\$NBK = 000004
\$.SAV = 010412R

WTIN = 000002
WTOU = 000001
XFLAG = 00005R
XMASK = 000021
XREG0 = 000200
XREG1 = 000220
XREG2 = 000240
XREG3 = 000260
XREG4 = 000300
XREG5 = 000340
XREG6 = 000360
ZCOND = 001400
\$NBK = 000004
\$.SAV = 010412R

WTIN = 000002
WTOU = 000001
XFLAG = 00005R
XMASK = 000021
XREG0 = 000200
XREG1 = 000220
XREG2 = 000240
XREG3 = 000260
XREG4 = 000300
XREG5 = 000340
XREG6 = 000360
ZCOND = 001400
\$NBK = 000004
\$.SAV = 010412R

WTIN = 000002
WTOU = 000001
XFLAG = 00005R
XMASK = 000021
XREG0 = 000200
XREG1 = 000220
XREG2 = 000240
XREG3 = 000260
XREG4 = 000300
XREG5 = 000340
XREG6 = 000360
ZCOND = 001400
\$NBK = 000004
\$.SAV = 010412R

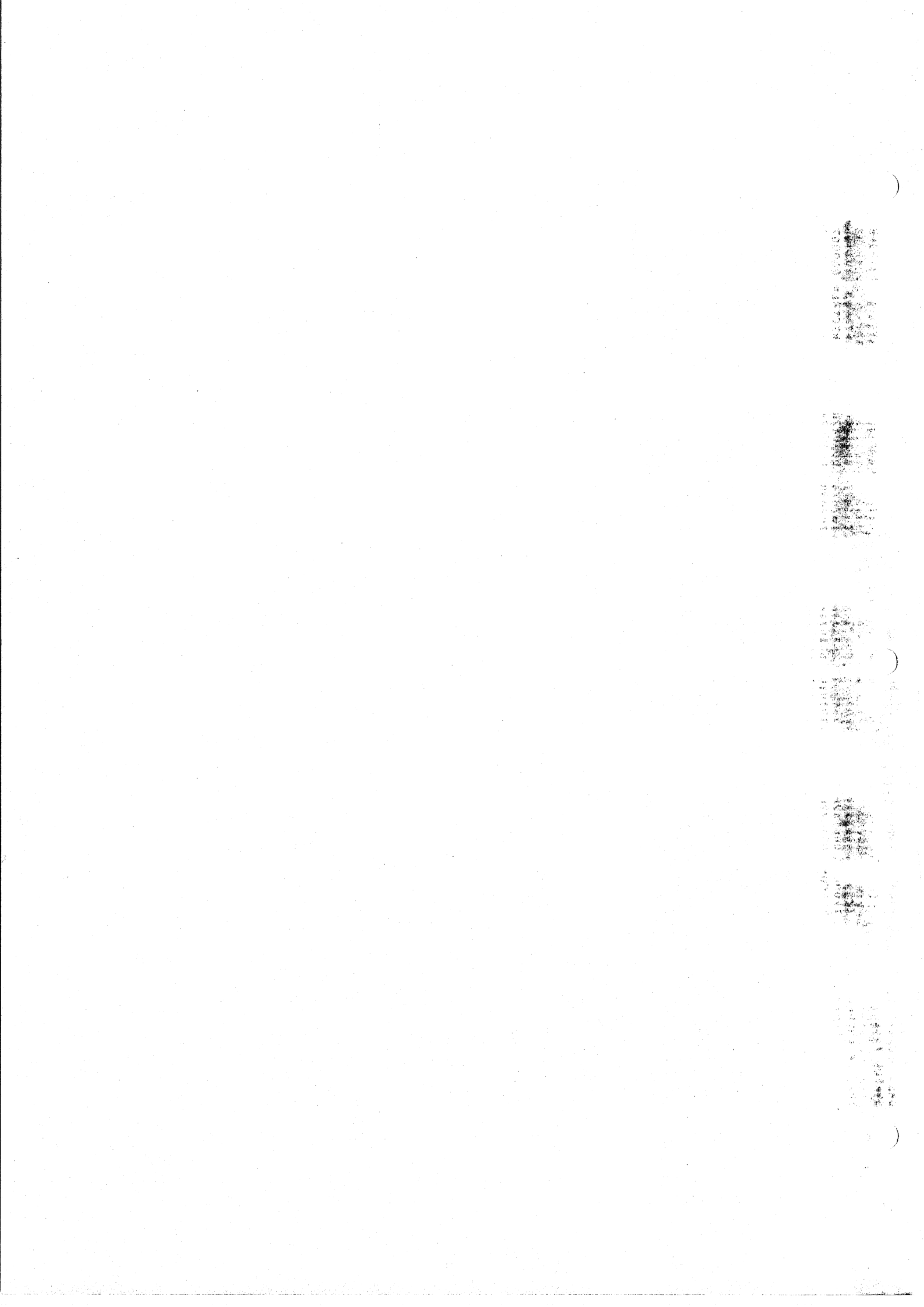
WTIN = 000002
WTOU = 000001
XFLAG = 00005R
XMASK = 000021
XREG0 = 000200
XREG1 = 000220
XREG2 = 000240
XREG3 = 000260
XREG4 = 000300
XREG5 = 000340
XREG6 = 000360
ZCOND = 001400
\$NBK = 000004
\$.SAV = 010412R

WTIN = 000002
WTOU = 000001
XFLAG = 00005R
XMASK = 000021
XREG0 = 000200
XREG1 = 000220
XREG2 = 000240
XREG3 = 000260
XREG4 = 000300
XREG5 = 000340
XREG6 = 000360
ZCOND = 001400
\$NBK = 000004
\$.SAV = 010412R

• ARS. 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

102020 001770H
102017 001752H
102016 001732H
102015 001702R
102014 001652F
102011 001604H
102005 001324E
102004 001320K
102003 001276R
102002 001230P
101014 001206K
101013 001170F
101012 001154K
101011 001124K
0004022H
0003326H
000350K
000354F
000300H
000420R
000460K
000510K
000546R
000572K
000610F
000644K
000660K
000672K
000676R



digital

Computer Special Systems