```
                .REM @


                        IDENTIFICATION
                        --------------

        PRODUCT CODE:       AC-E950B-MC

        PRODUCT NAME:       CXKMCB0 KMC-11 MODULE

        PRODUCT DATE:       SEPTEMBER 1978

        MAINTAINER:         DEC/X11 SUPPORT GROUP

        THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
        WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT
        BY DIGITAL EQUIPMENT CORPORATION.  DIGITAL EQUIPMENT
        CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT
        MAY APPEAR IN THIS MANUAL.

        THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE
        PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER
        SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS
        COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY
        OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

        DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR
        THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS
        NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C)  1976,1978 DIGITAL EQUIPMENT CORPORATION
```

```
                        1.      ABSTRACT

                                KMC IS AN IOMOD THAT EXERCISES UP TO AND INCLUDING TWO
                                CONSECUTIVELY ADDRESSED AND CONSECUTIVELY VECTORED KMC11
                                SYNCHRONOUS INTERFACES. IT USES NO LINE UNIT FOR RECEIVING
                                AND TRANSMITTING DATA. DATA BUFFERS ARE TRANSMITTED AND
                                RECEIVED FROM PDP11 MEMORY TO KMC11 & VICE VERSA.
                                 THE RECEIVER AND TRANSMITTER ISR'S ARE PERFORMED
                                AT LEVEL 0 (PIRQ) .DATA CHECKING IS PERFORMED AT LEVEL
                                0 AND DONE OUTSIDE THE ISRS.

                        2.      REQUIREMENTS

                                HARDWARE:       AT LEAST 1 KMC11

                                STORAGE:: KMC REQUIRES:
                                                1. DECIMAL WORDS: 2235
                                                2. OCTAL WORDS: 04273
                                                3. OCTAL BYTES: 10566


                        3.      PASS DEFINITION

                                ONE PASS OF THE KMAA MODULE CONSISTS OF TRANSMITTING AND
                                RECEIVING 1 BUFFERS OF 2-512 CHARACTERS 200 TIMES FOR EACH SELECTED
                                DEVICE.

                        4.      EXECUTION TIME

                                RUNNING ALONE ON AN 11/45 ONE PASS TAKES APPROXIMATELY
                                ONE MINUTE. IF RUN AT XX BAUD AND XX BUFFER SIZE

                        5.      CONFIGURATION PARAMETERS.

                                DEFAULT PARAMETERS:
                                ADDR: 1,  VECTOR: 1,  BR1: 5,  BR2: 5,  DVID1: 1,  SR1:0
                                KMAA WILL RUN UP TO TWO CONSECUTIVELY ADDRESSED AND
                                CONSECUTIVELY VECTORED KMC11'S.  THERE ARE THREE
                                PARAMETERS WHICH CAN BE CONTROLLED IN THIS MODULE.
                                1.NPR RATE:- THIS CONTROLS THE RATE OF NPR'S
                                OCCURING FROM KMC11'S. USING MODIFY COMMAND THIS CAN BE
                                SET TO SPECIFIC VALUE. THE ADDRESS OF THIS PARAMETER IS
                                226 IN KMAA MODULE.
                                THIS PARAMETER CAN BE CHOSEN IN TWO DIFFERENT WAYS.
                                I. WHEN SR1<BIT15>:=1 THEN WHATEVER IN LOC 222[RTMULV]
                                IS LOADED INTO NPRATE LOCATION[226].
                                II.WHEN SR1<BIT14>:=1 THEN NPRATE BECOMES EQUAL TO
                                RTMULV MULTIPLIED BY SR1<6:11>.
                                DEFAULT:: 10000[OCTAL]
                                RANGE:: 12-7888 USEC/NPR.

                                2.NPR/BR RATE:- CONTROLS THE RATE OF NPR'S PER INTERRUPT.
                                LIKE NPR RATE THIS PARAMETER CAN ALSO BE CHOSEN IN TWO
                                DIFFERENT WAYS. IN THIS CASE SZMULV[220] WILL BE USED
                                IN PLACE OF RTMULV. AND SR1<0:5> CONTENTS WILL BE USED TO MULTIPLY.
                                PARAMETERS :: RSIZE(202),XSIZE(204)
```

```
                        DEFAULT:: 377(OCTAL)

                        RANGE:: 1-377(OCTAL)
                        CAN BE SET USING MOD COMMAND
                        3.DIRECTION OF NPR'S:- THE DIRECTION OF NPR'S
                        CAN BE CHOSEN SETTING PROPER BIT IN SR1.
                        EXPLANATION FOLLOWS.


                6.      DEVICE/OPTION SETUP


                        SR1(SWITCH REGISTER CONTENTS)   OPTION.

                        BIT15:1 I.E SR1:10XXXX          NPRATE:= RTMULV * SR1 <6:11>
                                                        NPR/BR:= SZMULV * SR1 <0:5>
                        BIT14:1 I.E SR1:04XXXX          NPRATE:= RTMULV
                                                        NPR/BR:= SZMULV
                        BIT 15 & 14:0                   DEFAULT RATE.
                        BIT 15 & 14:1                   ILLEGAL.
                        BIT13:1 I.E SR1:X2XXXX          XMIT ONLY.
                        BIT12:1 I.E     SR1:X1XXXX                RECEIVE ONLY
                        BIT13 & 12:0                    DEFAULT.
                        BIT13 & 12:1                    ILLEGAL.
                        SR1 BITS 6:11                   NPR RATE MULTIPLIER.
                        SR1 BITS 5:0                    NPR/BR RATE MULTIPLIER.

                        NOTE:   SR1 CAN BE SET UP AT CONFIGURATION TIME OR
                                AT RUN TIME WITH A MOD COMMAND.
```

```
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
```

7. MODULE OPERATION

1. LOAD SOFTWARE POINTERS IN LINK TABLE. SET PARAMETERS.
2. LOAD VECTORS AND PRIORITIES IN TABLE
3. LOAD MICRO-CODE,VERIFY IT AND INITIATE IT.
4. ENABLE SELECTED DEVICES.
5. SCAN FOR ALL DEVICES TO FINISH
6. IF NOT DONE GO TO 4.
    IF HUNG REPORT SO AND DROP HUNG DEVICE.
7. CHECK DATA FOR ALL DEVICES SELECTED.
8. DECREMENT ITERATION COUNT
9. IF NOT = 0 GO TO 1
10. SIGNAL ENDPASS.

IISR: INPUT INTERRUPT SERVICE ROUTINE.
I1. GET INTERRUPTING KMCSCR.
I3. IF RECEIVE BA/CC WAS REQUESTED, LOAD REC BA/CC.
I4. IF XMIT BA/CC WAS REQUESTED, LOAD XMIT BA/CC.
15. RTI

OISR: OUTPUT INTERRUPT SERVICE ROUTINE.
O1. GET INTERRUPTING KMCSCR
O2. IF ERROR, REPORT IT AND EXIT.
O3. IF XMIT DONE OR REC DONE, SET APPROPRIATE BITS IN
    THE ENDPASS FLAG FOR THE DEVICE.
O4. RTI

9. NON-STANDARD PRINTOUTS

IF THE MODULE "HANGS" IN WHICH NOT ALL SELECTED DEVICES HAVE FINISHED, THEN A "HUNG" MESSAGE IS PRINTED OUT. CHECK THE ENDPASS FLAGS FOR EACH SELECTED DEVICE IN THE LINK TABLE TO DETERMINE WHICH DEVICE FAILED TO FINISH AND HOW FAR IT GOT.

FOR EXAMPLE:

THE TWO ENDPASS FLAGS ARE LOCATED IN THE LINK TABLE (INTLNK) AT THE FOLLOWING LOCATIONS.
XX11:
XX21:

ONLY BITS 0 THRU 3 ARE USED AND ARE DEFINED AS FOLLOWS:
BIT1 = 1 RECEIVE BA/CC'S WERE LOADED.
BIT0 = 1 TRANSMIT BA/CC'S WERE LOADED.
BIT2 = 1 TRANSMIT DONE'S WERE RECEIVED.
BIT3 = 1 RECEIVE DONE'S WERE RECEIVED.

A CORRECT END PASS FLAG = 17, WHEN THE ENDPASS FLAGS = 17 FOR THE SELECTED DEVICES, THE DATA IS CHECKED. IF A "HUNG" MESSAGE IS TYPED IT IS BECAUSE ONE OR BOTH DEVICES DID NOT FINISH. TO FIND WHICH ONE, CHECK THE END PASS FLAGS, ANY THAT ARE NOT EQUAL TO 17 ARE THE HUNG DEVICES. CHECK WHICH BITS OF THE ENDPASS FLAG ARE CLEAR TO SEE WHAT IT WAS TRYING TO DO.

@

```
                                                        IOMOD<KMCB >,1,1,5,5,0,200,136
                                        MODULE  140000,KMCB ,1,1,5,5,0,200,136
                                        .TITLE  KMCB DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6      23-MAY-78
                                        .LIST   BIN
                                ;*************************************************************************
000000'                         BEGIN:
000000' 046513  041103      040 MODNAM: .ASCII  /KMCB / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   OPEN        ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 000001                  ADDR:   1+0                 ;1ST DEVICE ADDR.
000010' 000001                  VECTOR: 1+0                 ;1ST DEVICE VECTOR.
000012'     240                 BR1:    .BYTE   PRTY5+0     ;1ST BR LEVEL.
000013'     240                 BR2:    .BYTE   PRTY5+0     ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1                 ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                ;SWITCH REGISTER 4
                                ;*************************************************************************
000026' 140000                  STAT:   140000              ;STATUS WORD.
000030' 000272'                 INIT:   START               ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP               ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                   ;PASS COUNTER.
000036' 000200                  ICONT:  200                 ;# OF ITERATIONS PER PASS=200
000040' 000000                  ICOUNT: 0                   ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0                   ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0                   ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0                   ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0                   ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0                   ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0                   ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                     ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0                   ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0                   ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN                ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN                ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN                ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN                ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN                ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN                ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN                ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN                ;ADDR OF CURRENT CSR.
000102'                         SBADR:                      ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                ;CONTENTS OF CSR.
000104'                         WASADR:                     ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                ;STATUS REG CONTENTS.
000106'                         ERRTYP:                     ;TYPE OF ERROR
000106' 000000                  ASD:    OPEN                ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                ;ACTUAL DATA.
000112' 000322'                 RSTRT:  RESTRT              ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                ;# OF INTERRUPTS PER ITERATION
000122' 000136                  IDNUM:  136                 ;MODULE IDENTIFICATION NUMBER=136
000224'                         MODSP:
```



```
                                ;*************************************************************************
                                ;
                                ;;
                                ;;*****************************************
                                ;
                                ;       VARIABLES FOR KMC11
                                ;
                                ;;*****************************************
                                ;;
                                ;
000224' 000000                  DLY1:   .WORD   0           ;DEVICE 1 DELAY COUNT.
000226' 000000                  DLY2:   .WORD   0           ;DEVICE 2 DELAY COUNT.
000230' 000000                  SELECT: .WORD   0           ;TEMPORARY SELECTED DEVICE'S
000232' 000017                  FLAGB:  .WORD   17          ;END PASS FLAG.
000234' 000000                  FIRST:  .WORD   0           ;FIRST PASS FLAG.
000236' 000000                  MASK:   .WORD   0           ;TEMPORARY VARIABLE.
000240' 000377                  RSIZE:  .WORD   377         ;RECEIVE BUFFER SIZE.
000242' 000377                  XSIZE:  .WORD   377         ;TRANSMIT BUFFER SIZE.
000244' 000000                  VA:     .WORD   0           ;VIRTUAL ADDRESS.
000246' 000000                  PA:     .WORD   0           ;PHYSICAL ADDRESS.
000250' 000000                  EA:     .WORD   0           ;EXTENDED ADDRESS.
000252' 000000                  SAR0:   .WORD   0           ;SAVE LOC.FOR R0.
000254' 000000                  SAR1:   .WORD   0           ;SAVE LOC. FOR R1
000256' 000002                  SZMULV: .WORD   2           ;LOCATION USED TO CALCULATE NPR/BR RATE.
000260' 000100                  RTMULV: .WORD   100         ;LOCATION USED TO CALCULATE NRR RATE.
                        000000  TERM=   0                   ;TERMINATING VALUE.
000262'     000                 RCOLV:  .BYTE   0           ;XMITR ONLY FLAG=SPAD<16>
000263'     000                 XMOLV:  .BYTE   0           ;RCV ONLY FLAG=SPAD <15>
                                        .EVEN
000264' 010000                  NPRTE:  .WORD   10000       ;LOCATION FOR NPR RATE.
000266' 000000                  TEMP:   .WORD   0           ;TEMPORARY VARIABLE.
000270' 000000                  FLAG:   .WORD   0           ;USED TO LOAD MAIN MEMORY.
                                ;;
                                ;;;
                                ;;;
                                ;;;:*****************************
                                ;;;
                                ;;;
                                ;;
                                ;
                                ;
                                ;
                                ;       BEGIN THE DEC./X11 MODULE FOR THE KMC11
                                ;
                                ;
                                ;

000272' 005067  177772          START:  CLR     FLAG        ;SET FOR FIRST PASS...
000276' 032767  177774  177510          BIT     #^C<3>,DVID1 ;DROP MODULE IF OTHER THEN
000304' 001004                          BNE     DROP        ;FIRST 2 DEVICES ARE SELECTED
000306' 016767  177502  177714          MOV     DVID1,SELECT ;SELECT=ACTIVE DEVICES.
000314' 001002                          BNE     RESTRT      ;DROP MODULE IF NO ACTIVE DEVICES.
000316'                         DROP:
000316' 104410  000000'                 ENDS,BEGIN          ;
000322' 005067  177706          RESTRT: CLR     FIRST       ;INITIALIZE THE FIRST TIME FLAG.
000326' 012700  005276'         LOOP:   MOV     #RBUF11,R0  ;GET SET TO CLEAR BUFFERS.
```

```
                                1$:     CLR     (R0)+           ;CLEAR THE LOCATION IN BUFFER.
000332' 005020                          CMP     R0,#BASE1       ;END OF BUFFERS?
000334' 020027  007276'                 BLE     1$              ;NO GO CLEAR THE NEXT...
000340' 003774                          MOV     SELECT,R0       ;R0=ACTIVE BITS.
000342' 016700  177662                  BEQ     DROP            ;DROP MODULE IF NO DEVICES SELECTED.
000346' 001763                          TST     FLAG            ;IS IT FIRST PASS???
000350' 005767  177714                  BNE     SETUP1          ;NO,THEN DONT LOAD THE MICRO-CODE...
000354' 001173
                                ;
                                ;:*****************************************************************
                                ;
                                ;       I.  THIS PART SETS UP PARAMETERS COMMON TO ALL
                                ;       DEVICES LIKE:   1)NPR/BR RATE.
                                ;                       2)NPR RATE.
                                ;                       3)DIRECTION OF NPR.
                                ;       II. INITIATES MICRO-CODE LOADED IN KMC11.
                                ;
                                ;:*****************************************************************
                                ;
000356' 105067  177700                  CLRB    RCOLY           ;INITIALIZE THE FLAGS.
000362' 105067  177675                  CLRB    XMOLY           ;INITIALIZE THE FLAGS.
000366' 005067  003642                  CLR     XX11            ;INITIALIZE THE END PASS FLAG.
000372' 005067  003660                  CLR     XX21            ;INITIALIZE THE END PASS FLAG.
000376' 005767  177414                  TST     SR1             ;IS IT MULTIPLY OPTION?
000402' 100416                          BMI     2$              ;YES,GO SET UP BFSIZE & NPRRATE.
000404' 032767  040000  177404          BIT     #BIT14,SR1      ;IS THE RATE SET?
000412' 001467                          BEQ     3$              ;NO,TAKE DEFAULT.
000414' 016767  177636  177616          MOV     SZMULV,RSIZE    ;SET NPR/BR RATE.
000422' 016767  177630  177612          MOV     SZMULV,XSIZE    ;SET THE NPR/BR RATE.PARAMETER.
000430' 016767  177624  177626          MOV     RTMULV,NPRTE    ;SET NPR RATE.
000436' 000455                          BR      3$              ;SET UP THE REST.
000440' 032767  040000  177350  2$:     BIT     #BIT14,SR1      ;IS SR1<14>.=1?
000446' 001061                          BNE     4$              ;YES, PRINT THE ERROR & DROP THE MODULE.
000450' 016701  177342                  MOV     SR1,R1          ;RETRIEVE BUFFER SIZE MULTIPLICAND.
000454' 016702  177336                  MOV     SR1,R2          ;RETRIEVE NPR RATE MULTIPLICAND.
000460' 006202                          ASR     R2              ;SET UP NPR RATE MULTIPLICAND
000462' 006202                          ASR     R2              ;SET UP NPR RATE MULTIPLICAND
000464' 006202                          ASR     R2              ;SET UP NPR RATE MULTIPLICAND
000466' 006202                          ASR     R2              ;SET UP NPR RATE MULTIPLICAND
000470' 006202                          ASR     R2              ;SET UP NPR RATE MULTIPLICAND
000472' 006202                          ASR     R2              ;SET UP NPR RATE MULTIPLICAND
000474' 042701  177700                  BIC     #177700,R1      ;CLEAR THE EXTRA BITS.
000500' 042702  177700                  BIC     #177700,R2      ;CLEAR UNNECESSARY BITS.
000504' 010103                          MOV     R1,R3           ;GET THE MULTIPLICAND...
000506' 016704  177544                  MOV     SZMULV,R4       ;GET THE MULTIPLIER...
000512' 004567  003466                  JSR     R5,MLTPLY       ;MULTIPLY & RETURN THE RESULT IN TEMP.
000516' 022767  000377  177542          CMP     #377,TEMP       ;CHECK IF WITHIN LIMIT...
000524' 003406                          BLE     12$             ;NO TAKE THE DEFAULT...
000526' 016767  177534  177504          MOV     TEMP,RSIZE      ;RSIZE=NPR/BR RATE.
000534' 016767  177526  177500          MOV     TEMP,XSIZE      ;XSIZE=NPR/BR RATE.
000542' 010203                  12$:     MOV     R2,R3           ;GET THE MULTIPLICAND...
000544' 016704  177510                  MOV     RTMULV,R4       ;GET THE MULTIPLIER...
000550' 004567  003430                  JSR     R5,MLTPLY       ;MULTIPLY & PUT RESULT IN TEMP.
000554' 022767  010000  177504          CMP     #10000,TEMP     ;CHECK IF WITHIN LIMIT...
000562' 003403                          BLE     3$              ;NO,TAKE THE DEFAULT...
000564' 016767  177476  177472          MOV     TEMP,NPRTE      ;NPRTE=NPR/TIME.
000572' 032767  010000  177216  3$:     BIT     #BIT12,SR1      ;IS IT RECEIVE ONLY?
```



```
000600' 001413                          BEQ     5$              ;NO , CHECK IF XMITR ONLY.
000602' 032767  020000  177206          BIT     #BIT13,SR1      ;IS  XMITR ONLY ALSO SET?
000610' 001404                          BEQ     6$              ;NO,SETUP FOR RECEIVE ONLY
000612'                         4$:
000612' 104403  000000' 002016'         MSGN$,BEGIN,SOFT1       ;ASCII MESSAGE CALL WITH COMMON HEADER
000620' 000636                          BR      DROP            ;DROP THE MODULE.
000622' 105167  177434          6$:     COMB    RCOLY           ;SET RECEIVE ONLY FLAG.
000626' 000406                          BR      7$              ;GO SET UP OTHER VARIABLES.
000630' 032767  020000  177160  5$:     BIT     #BIT13,SR1      ;IS XMITR ONLY SET?
000636' 001402                          BEQ     7$              ;NO,DO BOTH.
000640' 105167  177417                  COMB    XMOLY           ;SET XMITR ONLY FLAG.
000644' 016701  177136          7$:     MOV     ADDR,R1 ;GET THE FIRST DEVICE CSR.
000650' 016703  177354                  MOV     SELECT,R3       ;GET THE DEVICE SELECTED.
000654' 005067  177406          8$:     CLR     TEMP            ;CLEAR THE RETRY COUNT.
000660' 006203                          ASR     R3              ;ANY DEVICE REMAINS.
000662' 103404                          BCS     9$              ;YES,GO AND LOAD MICRO-CODE INTO IT.
000664' 001427                          BEQ     SETUP1          ;SETUP THE REST.
000666' 062701  000010          10$:    ADD     #10,R1          ;UPDATE THE CSR.
000672' 000770                          BR      8$              ;LOAD THE NEXT DEVICE.
000674' 004767  002456          9$:     JSR     PC,WCRAM        ;WRITE THE CRAM WITH MICRO-CODE.
000700' 004767  002646                  JSR     PC,WMEMRY       ;AND LOAD LOWER HALF OF MAIN MEMORY WITH XMITR BUFFER.
000704' 004767  002714                  JSR     PC,VERIFY       ;VERIFY MICRO-CODE & XMITR DATA.
000710' 005767  177322                  TST     MASK            ;IS THERE ANY ERROR.
000714' 001764                          BEQ     10$             ;NO,GO INITIATE IT.
000716' 104403  000000' 002022'         MSGN$,BEGIN,SOFT2       ;ASCII MESSAGE CALL WITH COMMON HEADER
000724' 005267  177336                  INC     TEMP            ;INCREMENT RETRY COUNT.
000730' 022767  000003  177330          CMP     #3,TEMP         ;IS IT TRIED THREE TIMES?
000736' 003356                          BGT     9$              ;NO,TRY AGAIN!!
000740' 000167  177352                  JMP     DROP            ;DROP THE MODULE.
                                ;
                                ;:*****************************************************
                                ;
                                ;       THIS PART SETSUP THE PROGRAM CONTROL
                                ;       VARIABLES FOR THE DEVICES AND THE
                                ;       MODULE. EX. QUEUES....,ETC.
                                ;
                                ;:*****************************************************
                                ;
000744' 016701  177036          SETUP1: MOV     ADDR,R1         ;GET THE DEVICE CSR.
000750' 016702  177034                  MOV     VECTOR,R2       ;GET THE VECTOR.
000754' 012703  004222'                 MOV     #INTLNK,R3      ;GET THE POINTER TO INTERRUPT LINKAGE.
000760' 016767  177246  003246          MOV     FLAG,XX11       ;SET THE END PASS FLAG FOR DEV#1.
000766' 016767  177240  003262          MOV     FLAG,XX21       ;SET THE END PASS FLAG FOR DEV#2.
000774' 012767  007276' 006354          MOV     #PIRINQ,INQIN   ;SET UP ALL QUEUES & ITS POINTERS.
001002' 012767  007276' 006350          MOV     #PIRINQ,INQOUT  ;SET UP ALL QUEUES & ITS POINTERS.
001010' 012767  007316' 006346          MOV     #PIROUTQ,OUTQOUT ;SET UP ALL QUEUES & ITS POINTERS.
001016' 012767  007316' 006336          MOV     #PIROUTQ,OUTQIN ;SET UP ALL QUEUES & ITS POINTERS.
001024' 012767  007336' 006334          MOV     #REGQ,REGQI     ;SET UP ALL QUEUES & ITS POINTERS.
001032' 012767  007336' 006330          MOV     #REGQ,REGQO     ;SET UP ALL QUEUES & ITS POINTERS.
001040' 016700  177164                  MOV     SELECT, R0      ;R0= DEVICES SELECTED.
001044' 006200                  2$:     ASR     R0              ;ANY DEVICE ACTIVE?
001046' 103410                          BCS     4$              ;YES, GO SET IT UP.
001050' 001446                          BEQ     SETUP2          ;ALL DONE?
001052' 062701  000010          3$:     ADD     #10,R1  ;NO,UPDATE CSR.
001056' 062702  000010                  ADD     #10,R2  ;UPDATE VECTOR
001062' 062703  000022                  ADD     #22,R3  ;UPDATE LINK.
```

```
001066' 000766                          BR      2$              ;GO SET UP FOR NEXT DEVICE.
001070' 010312                  4$:     MOV     R3,(R2)         ;LOAD INTERRUPT VECTOR ADDRESS.
001072' 116762  176714  000002          MOVB    BR1,2(R2)       ;SET THE PRIORITY.
001100' 010163  000010                  MOV     R1,10(R3)       ;SET THE DEVICE CSR.
001104' 010362  000004                  MOV     R3,4(R2)        ;LOAD XMITR
001110' 062762  000004  000004          ADD     #4,4(R2)        ;INTERRUPT VECTOR.
001116' 116762  176670  000006          MOVB    BR1,6(R2)       ;SET THE PRIORITY.
001124' 005063  000012                  CLR     12(R3)          ;CLEAR END PASS FLAG FOR DEV#1.
001130' 005063  000016                  CLR     16(R3)          ;CLEAR THE RECEIVE BUFFER OFFSET.
001134' 105767  177123                  TSTB    XMOLY           ;IS XMITR ONLY FLAG SET?
001140' 001404                          BEQ     5$              ;NO,CHECK FOR RECEIVE ONLY FLAG.
001142' 012763  000012  000012          MOV     #BIT1!BIT3,12(R3) ;SET XMITR BITS IN ENDPASS FLAG.
001150' 000740                          BR      3$              ;
001152' 105767  177104          5$:     TSTB    RCOLY           ;IS RECEIVE ONLY FLAG SET?
001156' 001735                          BEQ     3$              ;NO,DON'T SET ANY BITS IN ENDPASS FLAG.
001160' 012763  000005  000012          MOV     #BIT0!BIT2,12(R3) ;SET RECEIVE BITS IN ENDPASS FLAG.
001166' 016701  176614          SETUP2: MOV     ADDR,R1         ;R1=CSR ADDRESS.
001172' 016700  177032                  MOV     SELECT,R0       ;R0=SELECT.
001176' 006200                  1$:     ASR     R0              ;ANY DEVICE ACTIVE?
001200' 103404                          BCS     3$              ;YES,GO & INITIATE THE DEVICE DEC/X MODULE.
001202' 001454                          BEQ     SCAN            ;ALL DONE, GO AND SCAN.
001204' 062701  000010          2$:     ADD     #10,R1          ;UPDATE CSR ADDRESS.
001210' 000772                          BR      1$              ;CONTINUE.
001212' 005767  177016          3$:     TST     FIRST           ;IS IT FIRST PASS???
001216' 001013                          BNE     9$              ;NO,THEN DON'T INITIALIZE DEVICE
001220' 012711  040000                  MOV     #BIT14,(R1)     ;MASTER CLEAR FIRST TIME ONLY.
001224' 005011                          CLR     (R1)            ;INITIALIZE THE UNIBUS CSR'S.
001226' 005061  000002                  CLR     2(R1)           ;INITILIZE THE UNIBUS CSR'S.
001232' 005061  000004                  CLR     4(R1)           ;INITIALIZE THE UNIBUS CSR'S.
001236' 005061  000006                  CLR     6(R1)           ;INITIALIZE THE UNIBUS CSR'S.
001242' 012711  100000                  MOV     #BIT15,(R1)     ;SET THE RUN BIT...
001246' 105711                  9$:     TSTB    (R1)            ; IS RD I SET?
001250' 100415                          BMI     12$             ;YES, THEN START DECX...
001252' 010067  176774                  MOV     R0,SAR0         ;SAVE REGISTER R0...
001256' 010167  176772                  MOV     R1,SAR1         ;SAVE REGISTER R1...
001262' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001266' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001272' 016700  176754                  MOV     SAR0,R0         ;RESTORE REGISTER R0...
001276' 016701  176752                  MOV     SAR1,R1         ;RESTOREREGISTER R1...
001302' 000761                          BR      9$              ; WAIT FOR RD I TO SET...
001304' 052761  000020  000002  12$:    BIS     #020,2(R1)      ;SET IEI...
001312' 105767  176744                  TSTB    RCOLY           ;IS RECEIVE ONLY SET?
001316' 001403                          BEQ     6$              ;ONLY BRANCH IF NO.
001320' 052711  000020                  BIS     #020,(R1)       ;SET IEI, RECEIVE BA/CC I
001324' 000727                          BR      2$              ;CONTINUE.
001326' 052711  000024          6$:     BIS     #024,(R1)       ;SET IEI, XMITR BA/CC I.
001332' 000724                          BR      2$              ;CONTINUE.
                                ;
                                ;:
                                ;:**************************************************************
                                ;:*
                                ;:*     THIS ROUTINE SCANS ALL DEVICES END PASS FLAGS
                                ;:*     UNTIL ALL ACTIVE KMC11 DEVICES ARE FINISHED.
                                ;:*     UPDATES PASS COUNT AND LOOPS TILL 200 PASSES
                                ;:*     ARE DONE. CHECKS DATA AND PRINTS OUT DATA ERRORS.
                                ;:*     REPORTS END OF PASS.
```



```
                                ;:*
                                ;:**************************************************************
                                ;:
001334' 012767  000003  176674  SCAN:   MOV     #3,MASK         ;SET BIT FOR ALL DEVICES.
001342' 012767  000010  176654          MOV     #10,DLY1        ;SET DELAY COUNT.
001350' 005067  176652                  CLR     DLY2            ;CLEAR DELAY COUNT
001354' 026767  176652  002652  1$:     CMP     FLAGB,XX11      ;IS DEVICE 1 ALL DONE?.
001362' 001003                          BNE     2$              ;NO,CHECK THE NEXT ONE.
001364' 042767  000001  176644          BIC     #BIT0,MASK      ;CLEAR THE DEVICE BIT.
001372' 026767  176634  002656  2$:     CMP     FLAGB,XX21      ;IS DEVICE #2 ALL DONE?
001400' 001003                          BNE     3$              ;NO,GO AND WAIT.
001402' 042767  000002  176626          BIC     #BIT1,MASK      ;CLEAR THE DEVICE BIT.
001410' 005767  176622          3$:     TST     MASK            ;ARE ALL DEVICES DONE?
001414' 001064                          BNE     16$             ;NO,GO AND WAIT.
001416' 012701  004232'                 MOV     #INTLNK+10,R1   ;R1 POINTS TO DEVICE CSR.
001422' 016700  176602                  MOV     SELECT,R0       ;R0 CONTAINS BITS FOR ACTIVE DEVICES.
001426' 012703  005266'                 MOV     #BUFTAB,R3      ;R3=POINTER TO RECEIVER BUFFER.
001432' 006200                  4$:     ASR     R0              ;IS ANY DEVICE ACTIVE?
001434' 103417                          BCS     8$              ;YES,GO AND CHECK THE DATA.
001436' 001404                          BEQ     6$              ;IS IT ALL DONE THEN BRANCH.
001440' 062701  000022          5$:     ADD     #22,R1          ;UPDATE R1 TO NEXT DEVICE CSR.
001444' 005723                          TST     (R3)+           ;UPDATE R3 TO NEXT BUFFER.
001446' 000771                          BR      4$              ;CONTINUE.
001450' 012767  177777  176556  6$:     MOV     #-1,FIRST       ;SET FIRST PASS FLAG.
001456' 012767  177777  176604          MOV     #-1,FLAG        ;SET FLAG FOR MICRO-CODE LOADED.
001464' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
001470' 000167  176632          7$:     JMP     LOOP            ;LOOP THE MODULE.
                                ;
                                ;:**************************************************************
                                ;:*
                                ;:*     CHECK THE DATA IN RECEIVE BUFFER...
                                ;:*
                                ;:**************************************************************
                                ;
001474' 105767  176563          8$:     TSTB    XMOLY           ;IS IT XMIT ONLY???
001500' 001357                          BNE     5$              ;YES, THEN DONT CHECK THE DATA...
001502' 011302                          MOV     (R3),R2         ;R2=POINTS TO RECEIVER BUFFER.
001504' 012204                  9$:     MOV     (R2)+,R4        ;R4=POINTS TO RECEIVER DATA.
001506' 012705  004266'                 MOV     #XBUF,R5        ;R5=POINTS TO GOOD DATA.
001512' 016767  176522  176546          MOV     RSIZE,TEMP      ;SET THE BUFFER SIZE.
001520' 121514          10$:    CMPB    (R5),(R4)       ;COMPARE DATA.
001522' 001414                          BEQ     11$             ;GOOD,COMPARE NEXT CHAR.
001524' 011167  176350                  MOV     (R1),CSRA       ;LOAD CSRA.
001530' 010567  176346                  MOV     R5,SBADR        ;LOAD GOOD ADDRESS.
001534' 010467  176344                  MOV     R4,WASADR       ;LOAD BAD ADDRESS.
001540' 111567  176342                  MOVB    (R5),ASR        ;LOAD GOOD DATA.
001544' 111467  176340                  MOVB    (R4),AWAS       ;LOAD BAD DATA.
                                ;*******************************************************************
001550' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                ;*******************************************************************
001554' 122524          11$:    CMPB    (R5)+,(R4)+     ;POP BUFFER DATA POINTERS.
001556' 005367  176504                  DEC     TEMP            ;ALL DONE?
001562' 001356                          BNE     10$             ;NO, DO THE NEXT.
001564' 000725                          BR      5$              ;GO, AND CHECK REMAINING DEVICE.
001566'                         16$:
```

```
001566' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001572' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                                                               ;THEN CONTINUE AT NEXT INSTRUCTION.
001576' 005367  176424                 DEC     DLY2            ;DECREMENT DELAY COUNT FOR #2.
001602' 001402                         BEQ     .+6             ;
001604' 000167  177544                 JMP     1$              ;WAIT FOR DEVICE TO COMPLETE.
001610' 005367  176410                 DEC     DLY1            ;DECREMENT DELAY COUNT FOR #1.
001614' 001402                         BEQ     .+6             ;
001616' 000167  177532                 JMP     1$              ;WAIT FOR DEVICE TO COMPLETE.
001622' 016700  176410                 MOV     MASK,R0         ;R0=HUNG DEVICE BITS.
001626' 040067  176376                 BIC     R0,SELECT       ;DROP ANY HUNG DEVICE.
001632' 006000                         ROR     R0              ;WAS DEV#1 HUNG?.
001634' 103004                         BCC     17$             ;BRANCH IF NO.
001636' 004367  000024                 JSR     R3,XERR         ;TYPE ERROR MESSAGE & DROP.
001642' 004232'                        CSRG1                   ;POINTER TO DEV#1 CSR.
001644' 000001                         1                       ;DEVICE NUMBER FOR TYPEOUT.
001646' 006000                 17$:    ROR     R0              ;WAS DEV#2 HUNG?
001650' 103004                         BCC     18$             ;BRANCH IF NOT.
001652' 004367  000010                 JSR     R3,XERR         ;TYPE ERROR MESSAGE THEN DROP.
001656' 004254'                        CSR22                   ;POINTER TO DEV#2 CSR.
001660' 000002                         2                       ;DEVICE NUMBER FOR TYPEOUT.
001662' 000167  176440         18$:    JMP     LOOP            ;RESTART MODULE.
                               ;
                               ;;*********************************************************
                               ;;*
                               ;;*     THIS SUBROUTINE DROPS THE HUNG DEVICE.
                               ;;*     CLEARS OUT DEVICE SELECT BITS.
                               ;;*     PRINTS THE EXTENDED ERROR MESSAGE.
                               ;;*
                               ;;*********************************************************
                               ;
001666' 012302                 XERR:   MOV     (R3)+,R2        ;GET POINTER TO CSR.
001670' 012367  000110                 MOV     (R3)+,DEV       ;GET DEVICE #.
001674' 052767  000060  000102         BIS     #60,DEV         ;MAKE IT ASCII.
001702' 104403  000000' 002006'        MSGNS,BEGIN,DROP1       ;ASCII MESSAGE CALL WITH COMMON HEADER
001710' 011201                         MOV     (R2),R1         ;GET CSR ADDRESS.
001712' 010167  176162                 MOV     R1,CSRA         ;SAVE CSR
001716' 011167  176160                 MOV     (R1),ACSR       ;SAVE CONTENTS OF SEL0.
001722' 016167  000002  176154         MOV     2(R1),ASTAT     ;SAVE CONTENTS OF SEL2.
001730' 016167  000004  176266         MOV     4(R1),DLY1      ;SAVE CONTENTS OF SEL4.
001736' 016167  000006  176262         MOV     6(R1),DLY2      ;SAVE CONTENTS OF SEL6.
001744' 016267  000002  000274         MOV     2(R2),ESAV1     ;SAVE END PASS FLAG.
001752' 011267  000272                 MOV     (R2),ESAV2      ;SAVE RECEIVE BUFFER OFFSET.
001756' 016267  000010  000266         MOV     10(R2),ESAV3    ;SAVE RECV/XMITR COUNTERS.
001764' 012767  000011  176114         MOV     #11,ERRTYP      ;NO INTERRUPT
                                       ;*********************************************************
001772' 104405  000000' 002256'        $RDERS,BEGIN,ETABLE     ;DUMP KMC CSR'S AND STATUS FLAGS
                                       ;*********************************************************
002000' 005011                         CLR     (R1)            ;SHUT OFF HUNG KMC11.
002002' 000203                         RTS     R3
002004' 000000                 DEV:    .WORD   0
002006' 002026'                DROP1:  XDROP1
002010' 002004'                        DEV
002012' 002050'                        XDROP2
002014' 177777                         -1
```



```
002016' 002110'                SOFT1:  SOFT11
002020' 177777                         -1
002022' 002171'                SOFT2:  SOFT21
002024' 177777                         -1
                               ;;*
                               ;;*     ERROR MESSAGES..........
                               ;;*

002026' 020045  046513  030503 XDROP1: .ASCIZ  /% KMC11 DEVICE # /
002034' 020061  042504  044526
002042' 042503  021440  000040

002050' 051511  044040  047125 XDROP2: .ASCIZ  /IS HUNG AND HAS BEEN DROPPED..%/
002056' 020107  047101  020104
002064' 040510  020123  042502
002072' 047105  042040  047522
002100' 050120  042105  027056
002106' 000045

002110' 020045  051105  047522 SOFT11: .ASCIZ  /% ERROR IN SETTING UP SWITCH REGISTER % RESTART /
002116' 020122  047111  051440
002124' 052105  044524  043516
002132' 052440  020120  053523
002140' 052111  044103  051040
002146' 043505  051511  042524
002154' 020122  020045  042522
002162' 052123  051101  020124
002170'     000

002171'     045 042440  051122 SOFT21: .ASCIZ  /% ERROR IN LOADING MICRO-CODE WILL RETRY...%/
002176' 051117  044440  020116
002204' 047514  042101  047111
002212' 020107  044515  051103
002220' 026517  047503  042504
002226' 053440  046111  020114
002234' 042522  051124  027131
002242' 027056  000045

                                       .EVEN
                               ;;*
                               ;;*     EXTENDED ERROR PRINTOUT LOCATIONS....
                               ;;*
002246' 000000                 ESAV1:  .WORD   0
002250' 000000                 ESAV2:  .WORD   0
002252' 000000                 ESAV3:  .WORD   0
002254' 000000                 ESAV4:  .WORD   0
                               ;;*
                               ;;*     TABLE OF ADDRESSES FOR EXTERNAL ERROR PRINT OUTS....
                               ;;*
002256' 000224'                ETABLE: DLY1
002260' 000226'                        DLY2
002262' 002246'                        ESAV1
002264' 002250'                        ESAV2
002266' 002252'                        ESAV3
002270' 177777                         -1
002272' 002246'                FTABLE: ESAV1
002274' 002250'                        ESAV2
```

```
002276' 002252'                         ESAV3
002300' 002254'                         ESAV4
002302' 177777                          -1


                                ;;
                                ;;*******************************************************
                                ;;*
                                ;;*     INPUT INTERRUPT SERVICE ROUTINE
                                ;;*     THIS ROUTINE SERVES THE IN INTERRUPT
                                ;;*     FROM KMC11 BY LOADING REQUESTED XMITR
                                ;;*     OR RECEIVE BA/CC I
                                ;;*
                                ;;*******************************************************
                                ;;
002304' 010577  005046          INISR:  MOV     R5,@INQIN       ;SAVE LINK POINTER IN QUEUE.
002310' 062767  000002  005040          ADD     #2,INQIN        ;UPDATE THE QUEUE POINTER.
002316' 022767  007316' 005032          CMP     #PIRING+20,INQIN ;END OF QUEUE?
002324' 003003                          BGT     1$              ;NO,PROCEED.
002326' 012767  007276' 005022          MOV     #PIRING,INQIN   ;RESET QUEUE POINTER.
002334' 012605                  1$:     MOV     (SP)+,R5        ;RESTORE R5,I.E POP THE STACK.
                                        ;-------------------------------------------------------------------
002336' 000004  000000' 002344'         PIRQ$,BEGIN,2$          ; QUEUE UP TO CONTINUE AT 2$ AND RTI
                                        ;-------------------------------------------------------------------
002344' 017705  005010          2$:     MOV     @INQOUT,R5      ;RESTORE THE LINK POINTER.
002350' 062767  000002  005002          ADD     #2,INQOUT       ;UPDATE THE QUEUE POINTER.
002356' 022767  007316' 004774          CMP     #PIRING+20,INQOUT ;END OF QUEUE?
002364' 003003                          BGT     3$              ;NO,CONTINUE.
002366' 012767  007276' 004764          MOV     #PIRING,INQOUT  ;RESET THE QUEUE POINTER.
002374' 016501  000004          3$:     MOV     4(R5),R1        ;LOAD CSR ADDRESS.
002400' 032711  000007                  BIT     #7,(R1)         ;RECEIVE BA/CC1.?
002404' 001410                          BEQ     RECV            ;BRANCH IF YES.
002406' 032711  000004                  BIT     #4,(R1)         ;XMITR BA/CC I?
002412' 001033                          BNE     XMITR           ;BRANCH IF YES.
002414' 104403  000000' 002554'         MSGN$,BEGIN,ILINT       ;ASCII MESSAGE CALL WITH COMMON HEADER
002422' 104400  000000'                 EXIT$,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002426' 016504  000010          RECV:   MOV     10(R5),R4       ;GET RECEIVE BUFFER POINTER.
002432' 011467  175606                  MOV     (R4),VA         ;VA=RECEIVE BUFFER VIRTUAL ADDRESS..
002436' 004767  001510                  JSR     PC,EABITS       ;GET PHYSICAL ADDRESS.
002442' 016761  175600  000004          MOV     PA,4(R1)        ;SEL4=PHYSICAL RBUFF.
002450' 016761  175574  000006          MOV     EA,6(R1)        ;SEL6=EXTENDED BITS OF ADDRESS.
002456' 056761  175556  000006          BIS     RSIZE,6(R1)     ;LOAD RECEIVE CHARACTER COUNT.
002464' 052765  000002  000006          BIS     #BIT1,6(R5)     ;SET BIT1 IN END PASS FLAG FOR
                                                                ;REC BA/CC I LOADED.
002472' 142711  000220                  BICB    #BIT7!BIT4,(R1) ;CLEAR RDI TO REQUEST SERVICE FROM
                                                                ;KMC11.  CLEAR IEI...
002476' 104400  000000'                 EXIT$,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002502' 012767  004266' 175534  XMITR: MOV     #XBUF,VA        ;LOAD XBUFFER VIRTUAL ADDRESS.
002510' 004767  001436                  JSR     PC,EABITS       ;GET EXTENDED PHYSICAL ADDRESS.
002514' 016761  175526  000004          MOV     PA,4(R1)        ;LOAD PHYSICAL XMITR BUFFER.
002522' 016761  175522  000006          MOV     EA,6(R1)        ;LOAD EXTENDED ADDRESS BITS XMITR BUFFER.
002530' 056761  175506  000006          BIS     XSIZE,6(R1)     ;LOAD XMITR CHARACTER COUNT.
002536' 052765  000001  000006          BIS     #BIT0,6(R5)     ;SET BIT0 IN END PASS FLAG FOR
                                                                ;XMITR BA/CC I LOADED.
002544' 142711  000200          1$:     BICB    #BIT7,(R1)      ;CLEAR RD I TO INDICATE START OF OPERATION
002550' 104400  000000'                 EXIT$,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002554' 002560'                 ILINT:  MES2
```



```
002556' 177777                          -1
002560' 044445  050116  052125  MES2:   .ASCIZ /%INPUT INTERRUPT WITH NO REQUEST SET UP!!!%/
002566' 044440  052116  051105
002574' 052522  052120  053440
002602' 052111  020110  047516
002610' 051040  050505  042525
002616' 052123  051440  052105
002624' 052440  020520  020441
002632' 000045

                                        .EVEN
                                ;
                                ;;*******************************************************
                                ;;*
                                ;;*     OUTPUT INTERRUPT SERVICE ROUTINE
                                ;;*     THIS ROUTINE SERVES THE OUT INTERRUPT
                                ;;*     FROM KMC11 BY 1 REPORTING ERROR.
                                ;;*             2 ACCEPTING RECEIVE OR XMITR DONE
                                ;;*
                                ;;*******************************************************
                                ;
002634' 010577  004522          OUISR:  MOV     R5,@OUTQIN      ;SAVE LINK POINTER IN QUEUE.
002640' 062767  000002  004514          ADD     #2,OUTQIN       ;UPDATE THE QUEUE POINTER.
002646' 022767  007336' 004506          CMP     #PIROUTQ+20,OUTQIN ;IS IT END OF QUEUE?
002654' 003003                          BGT     1$              ;NO, CONTINUE.
002656' 012767  007316' 004476          MOV     #PIROUTQ,OUTQIN ;RESET THE QUEUE POINTER.
002664' 012605                  1$:     MOV     (SP)+,R5        ;POP THE STACK POINTER
                                        ;-------------------------------------------------------------------
002666' 000004  000000' 002674'         PIRQ$,BEGIN,2$          ; QUEUE UP TO CONTINUE AT 2$ AND RTI
                                        ;-------------------------------------------------------------------
002674' 017705  004464          2$:     MOV     @OUTQOUT,R5     ;RESTORE THE LINK POINTER FROM QUEUE.
002700' 062767  000002  004456          ADD     #2,OUTQOUT      ;UPDATE THE QUEUE POINTER.
002706' 022767  007336' 004450          CMP     #PIROUTQ+20,OUTQOUT ;IS IT END OF QUEUE?
002714' 003003                          BGT     3$              ;NO, CONTINUE.
002716' 012767  007316' 004440          MOV     #PIROUTQ,OUTQOUT ;RESET THE QUEUE POINTER.
002724' 011501                  3$:     MOV     (R5),R1         ;LOAD CSR ADDRESS.
002726' 032761  000002  000002          BIT     #BIT1,2(R1)     ;IS CONTROL/O ERROR REPORT?
002734' 001450                          BEQ     4$              ;NO, THEN CHECK FOR RCV OR XMITR.
002736' 032761  000010  000002          BIT     #BIT3,2(R1)     ;IS IT SOFT ERROR...
002744' 001403                          BEQ     9$              ;NO,THEN CHECK REMAINING...
002746' 104403  000000' 003156'         MSGN$,BEGIN,SFT1        ;ASCII MESSAGE CALL WITH COMMON HEADER
002754' 032761  000100  000002  9$:     BIT     #BIT6,2(R1)     ;IS IT DATA ERROR...
002762' 001403                          BEQ     12$             ;NO,CHECK THE REMAINING...
002764' 104403  000000' 003162'         MSGN$,BEGIN,DTER1       ;ASCII MESSAGE CALL WITH COMMON HEADER
002772' 032761  000040  000002  12$:    BIT     #BIT5,2(R1)     ;IS IT NON EX MEMORY ERROR.
003000' 001403                          BEQ     5$              ;NO THEN MUST BE DATA ERROR.
003002' 104403  000000' 003152'         MSGN$,BEGIN,NXMMRY      ;ASCII MESSAGE CALL WITH COMMON HEADER
003010' 010167  175064          5$:     MOV     R1,CSRA         ;LOAD DEVICE CSR ADDRESS.
003014' 016167  000004  175060          MOV     4(R1),ACSR      ;LOAD CONTENTS OF DEVICE CSR.
003022' 016167  000006  175054          MOV     6(R1),ASTAT     ;LOAD ERROR BITS.
003030' 012767  000001  175050          MOV     #1,ERRTYP       ;DATA ERROR
                                        ;*******************************************************************
003036' 104405  000000' 000000          ERDER$,BEGIN,NULL       ;A CNTL/O RECEIVED
                                        ;*******************************************************************
                                                                ;ASTAT= ERROR BITS.
003044' 142761  000352  000002          BICB    #352,2(R1)      ;CLEAR RDO,NON EX MEM.,CNTL/O.
```

```
                                        EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                                                        ;
003056' 032761  000004  000002  4$:     BIT     #BIT2,2(R1)             ;IS IT  XMITR DONE?
003064' 001411                          BEQ     6$                      ;NO,RECEIVE!SERVE IT.
                                                                        ;
003066' 052765  000004  000002          BIS     #BIT2,2(R5)             ;SET XMITR DONE BIT IN END PASS FLAG.
003074' 105767  175163                  TSTB    XMOLY                   ;IS IT XMIT ONLY???
003100' 001006                          BNE     7$                      ;CONTINUE.....
003102' 142711  000004                  BICB    #4,(R1)                 ;SET FOR RECEIVE BA/CC I...
003106' 000403                          BR      7$                      ;CONTINUE.
003110' 052765  000010  000002  6$:     BIS     #BIT3,2(R5)             ;SET RECEIVE DONE BIT IN ENDPASS FLAG.
003116' 026765  175110  000002  7$:     CMP     FLAGR,2(R5)             ;ALL DONE?
003124' 001005                          BNE     8$                      ;NO,CONTINUE.
003126' 142761  000020  000002          BICB    #BIT4,2(R1)             ;CLEAR IEO.
003134' 142711  000020                  BICB    #BIT4,(R1)              ;CLEAR THE IEI.
003140' 142761  000205  000002  8$:     BICB    #BIT7!BIT2!BIT0,2(R1)   ;CLEAR RDO,XMITR OR RECV DONE, BA/CC 0...
003146' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003152' 003166'                 NXMMRY: MES1
003154' 177777                          -1
003156' 003303'                 SFT1:   MES4
003160' 177777                          -1
003162' 003322'                 DTER1:  MES5
003164' 177777                          -1
003166' 020045  047516  020116  MES1:   .ASCII  /% NON EXISTENT MEMORY ADDRESS ERROR  %/
003174' 054105  051511  042524
003202' 052116  046440  046505
003210' 051117  020131  042101
003216' 051104  051505  020123
003224' 051105  047522  020122
003232' 022440
003234' 020040  041517  052503          .ASCIZ  /  OCCURED WHILE DOING NPR'S..... %/
003242' 042522  020104  044127
003250' 046111  020105  047504
003256' 047111  020107  050116
003264' 023522  027123  027056
003272' 027056  027056  027056
003300' 022440      000
003303'     045 051440  043117  MES4:   .ASCIZ  /% SOFT ERROR %/
003310' 020124  051105  047522
003316' 020122  000045
003322' 020045  040504  040524  MES5:   .ASCIZ  /% DATA ERROR ON TRANSMIT %/
003330' 042440  051122  051117
003336' 047440  020116  051124
003344' 047101  046523  052111
003352' 022440      000
                003356'                 .EVEN
                                        ;
                                        ;:*************************************************************
                                        ;:*
                                        ;:*     SUBROUTINE TO LOAD MICRO-CODE INTO CRAM.
                                        ;:*     AND THE PARAMETERS INTO SCRATCH PAD.
                                        ;:*     R1 CONTAINS THE CSR AT THE TIME OF ENTRY.
                                        ;:*
                                        ;:*************************************************************
                                        ;
003356' 005000                  WCRAM:  CLR     R0                      ;R0=POINTS TC CRAM ADDRESS.
```



```
003360' 012702  007372'                 MOV     #KMAAMC,R2              ;R2 POINTS TO MICRO-CODE.
003364' 005011                  1$:     CLR     (R1)                    ;CLEAR SEL0...
003366' 010061  000004                  MOV     R0,4(R1)                ;LOAD THE CRAM ADDRESS...
003372' 012261  000006                  MOV     (R2)+,6(R1)             ;LOAD WORD TC BE WRITTEN...
003376' 012711  002000                  MOV     #BIT10,(R1)             ;SET ROMO
003402' 012711  022000                  MOV     #BIT13!BIT10,(R1)               ;WRITE IT!
003406' 005200                          INC     R0                      ;INCREMENT CRAM ADDRESS.
003410' 022700  002000                  CMP     #2000,P0                ;OVER FLOW?
003414' 002430                          BLT     2$                      ;YES,REPORT AND RETURN.
003416' 022712  000000                  CMP     #0,(R2)                 ;IS IT END OF MICRO-CODE?
003422' 001360                          BNE     1$                      ;NO,CONTINUE LOADING.
003424' 012705  000011                  MOV     #11,R5                  ;R5=THE SCRATCH PAD ADDRESS
003430' 012702  000262'                 MOV     #RCOLV,R2               ;SET THE PARAMETER POINTER.
003434' 005011                  3$:     CLR     (R1)                    ;CLEAR SEL0.
003436' 042767  000077  000014          BIC     #77,24$                 ;CLEAR THE ADDRESS IN INSTRUCTION..
003444' 050567  000010                  BIS     R5,4$                   ;SET SCRATCH PAD ADDRESS
003450' 112261  000004                  MOVB    (R2)+,4(R1)             ;LOAD SEL4 WITH DATA.
003454' 004367  000446                  JSR     R3,ROMCLK               ;CLOCK THE INSTRUCTION IN
003460' 123100                  4$:     123100                          ; THIS LOCATION.
003462' 005205                          INC     R5                      ;INCREMENT SCRATCH PAD ADDRESS.
003464' 022705  000015                  CMP     #15,R5                  ;IS IT DONE?
003470' 003361                          BGT     3$                      ;BRANCH IF NOT DONE.
003472' 005011                  5$:     CLR     (R1)                    ;CLEAR SEL0.
003474' 000207                  2$:     RTS     PC                      ;RETURN
003476'
003476' 104403  000000' 003506'         MSGN$,BEGIN,CRMOFW              ;ASCII MESSAGE CALL WITH COMMON HEADER
003504' 000772                          BR      5$                      ;RETURN.

003506' 003512'                 CRMOFW: MES3
003510' 177777                          -1
003512' 020045  044515  051103  MES3:   .ASCIZ  /% MICRO-CODE OVER FLOWS CRAM %/
003520' 026517  047503  042504
003526' 047440  042526  020122
003534' 046106  053517  020123
003542' 051103  046501  022440
003550'     000
                003552'                 .EVEN
                                        ;
                                        ;:*****************************************************************
                                        ;:*
                                        ;:*     THIS ROUTINE WRITES GOOD DATA BUFFER
                                        ;:*     INTO LOWER HALF OF THE MAIN MEMORY...
                                        ;:*     R1 CONTAINS THE DEVICE CSR ADDRESS.
                                        ;:*
                                        ;:*****************************************************************
                                        ;
003552'                         WMEMRY:
003552' 004367  000350                  JSR     R3,ROMCLK               ;CLOCK INSTRUCTION.
003556' 010000                          010000                  ;LOAD MAR<0:17>.
003560' 004367  000342                  JSR     R3,ROMCLK               ;CLOCK INSTRUCTION.
003564' 004002                          004002                  ;SET POINTER IN KMC11 MEMORY.
003566' 012700  004266'                 MOV     #XBUF,R0                ;SET POINTER TO DATA.
003572' 012702  001000                  MOV     #1000,R2                ;SET THE COUNT.
003576' 005061  000004                  CLR     4(R1)                   ;
003602' 112061  000004          1$:     MOVB    (R0)+,4(R1)             ;BSEL4<--GOOD DATA.
003606' 004367  000314                  JSR     R3,ROMCLK               ;
```

```
                                                136500          ;LOAD MEMORY AND INCREMENT ADDRESS.
003612' 136500                                  DEC     R2              ;COUNT BY ONE.
003614' 005302                                  BNE     1$              ;BRANCH IF NOT DONE..
003616' 001371                                  CLR     (R1)            ;CLEAR THE CSR.
003620' 005011                                  RTS     PC              ;RETURN TO MAINLAND.
003622' 000207
                                        ;
                                        ;:*************************************************
                                        ;:*
                                        ;:*     VERIFY THE MICRO-CODE
                                        ;:*     THIS ROUTINE VERIFIES THE MICRO-CODE
                                        ;:*     LOADED. AND CHECKS NPR RATE PARAMETER.
                                        ;:*     AND THE MAIN MEMORY CONTENTS.....
                                        ;:*     R1 CONTAINS THE ADDRESS OF DEVICE CSR.
                                        ;:*
                                        ;:*************************************************
                                        ;
003624'                                 VERIFY:
003624' 005067  174406                          CLR     MASK            ;CLEAR THE ERROR FLAG.
003630' 012700  007372'                         MOV     #KMAAMC,R0      ;R0 POINTS TO SOFTWARE MICRO-CODE.
003634' 005002                                  CLR     R2              ;R2 CONTAINS CRAM ADDRESS BITS 0-7
003636' 005011                          1$:     CLR     (R1)            ;CLEAR THE MAINTANENCE REGISTER..
003640' 010261  000004                          MOV     R2,4(R1)        ;SET THE CRAM ADDRESS...
003644' 012711  002000                          MOV     #BIT10,(R1)     ;SET ROMD.
003650' 011005                                  MOV     (R0),R5         ;PUT "EXPECTED" IN R5.
003652' 016104  000006                          MOV     6(R1),R4        ;PUT "FOUND" IN R4.
003656' 020504                                  CMP     R5,R4           ;COMPARE
003660' 001403                                  BEQ     3$              ;BRANCH IF O.K.
003662' 005167  174350                          COM     MASK            ;SET ERROR FLAG.
003666' 000515                                  BR      4$              ;RETURN.
003670' 005720                          3$:     TST     (R0)+           ;BUMP MICRO-CODE POINTER.
003672' 005202                                  INC     R2              ;BUMP CRAM ADDRESS.
003674' 022710  000000                          CMP     #0,(R0)         ;IS IT DONE?
003700' 001403                                  BEQ     5$              ;YES,GO AND CHECK THE MEMORY...
003702' 022702  002000                          CMP     #2000,R2        ;DONE YET???
003706' 002353                                  BGE     1$              ;NO,CONTINUE......
003710'                                 5$:                             ;
003710' 012705  000011                          MOV     #11,R5          ;R5=SPAD ADDRESS.
003714' 012704  000262'                         MOV     #RCOLY,R4       ;SET THE PARAMETER ADDRESS.
003720' 005011                          6$:     CLR     (R1)            ;CLEAR SEL0.
003722' 042767  000077  000014                  BIC     #77,7$          ;CLEAR THE ADDRESS FIELD...
003730' 050567  000010                          BIS     R5,7$           ;SET SCRATCH PAD ADDRESS IN INSTR.
003734' 005061  000004                          CLR     4(R1)           ;CLEAR SEL4
003740' 004367  000162                          JSR     R3,ROMCLK       ;CLOCK THE INSTRUCTION.
003744' 040600                          7$:     040600                  ;MOVE   SPAD<R5>->SPERG.
003746' 004367  000154                          JSR     R3,ROMCLK       ;CLOCK THE INSTRUCTION IN NEXT LOCATION.
003752' 061224                                  061224                  ;MOVE BREG,OUT1<CSR4>.
003754' 122461  000004                          CMPB    (R4)+,4(R1)     ;COMPARE
003760' 001403                                  BEQ     8$              ;GOOD, DO THE NEXT.
003762' 005167  174250                          COM     MASK            ;SET ERROR FLAG.
003766' 000455                                  BR      4$              ;RETURN.
003770' 005205                          8$:     INC     R5              ;UPDATE R5 POINTING TO NEXT SPAD.
003772' 022705  000015                          CMP     #15,R5          ;IS IT DONE?
003776' 003350                                  BGT     6$              ;BRANCH IF NOT DONE.
004000' 012700  004266'                 9$:     MOV     #XBUF,R0        ;GET THE DATA ADDRESS.
004004' 012767  001000  174256                  MOV     #1000,FLAG      ;SET THE BUFFER POINTER.
004012' 016702  174224                          MOV     XSIZE,R2        ;SET THE COUNT...
```



```
004016' 042767  000377  000026  12$:    BIC     #377,15$        ;CLEAR THE ADDRESS FIELD.
004024' 042767  000003  000026           BIC     #3,18$          ;CLEAR THE ADDRESS FIELD.
004032' 156767  174232  000012           BISB    FLAG,15$        ;ADD ADDRESS TO INSTRUCTION.
004040' 156767  174225  000012           BISB    FLAG+1,18$      ;ADD ADDRESS TO INSTRUCTION.
004046' 004367  000054                   JSR     R3,ROMCLK       ;
004052' 010000                   15$:    010000                  ;LOAD MAR_LO.
004054' 004367  000046                   JSR     R3,ROMCLK       ;
004060' 004000                   18$:    004000                  ;LOAD MAR_HI.
004062' 004367  000040                   JSR     R3,ROMCLK       ;
004066' 040620                           040620                  ;BREG<--MEM.
004070' 004367  000032                   JSR     R3,ROMCLK       ;
004074' 061224                           061224                  ;RSEL4<--BREG.
004076' 122061  000004                   CMPB    (R0)+,4(R1)     ;COMPARE THE DATA.
004102' 001403                           BEQ     21$             ;BRANCH IF GOOD.
004104' 005167  174126                   COM     MASK            ;ELSE,SET THE ERROR FLAG.
004110' 000404                           BR      4$              ;RETURN.....
004112' 005267  174152           21$:    INC     FLAG            ;INCREMENT THE ADDRESS...
004116' 005302                           DEC     R2              ;DONE?
004120' 001336                           BNE     12$             ;NO,CHECK THE NEXT.
004122' 005011                   4$:     CLR     (R1)            ;CLEAR SEL0.
004124' 000207                           RTS     PC              ;RETURN.
                                        ;
004126' 112761  001000  000001  ROMCLK:  MOVB    #BIT9,1(R1)     ;SET ROMI
004134' 012361  000006                   MOV     (R3)+,6(R1)     ;LOAD INSTRUCTION IN SEL6.
004140' 052711  001400                   BIS     #BIT9!BIT8,(R1) ;CLOCK INSTRUCTION.
004144' 042711  003400                   BIC     #BIT10!BIT9!BIT8,(R1) ;CLEAR ROMD,ROMI,NP STEP
004150' 000203                           RTS     R3              ;RETURN.
                                        ;
                                        ;:*****************************************************************
                                        ;:*
                                        ;:*     SUBROUTINE EABITS:- GETS PHYSICAL 18 BITS ADDRESS FOR
                                        ;:*     16 BITS VIRTUAL ADDRESS.
                                        ;:*     RETURNS: ADRESS IN  PA::ADRESS<0:15>
                                        ;:*                         EA::ADDRESS<16:17>
                                        ;:*
                                        ;:*****************************************************************
                                        ;
004152'                                 EABITS:
004152' 104415  000000' 000244'                 GETPA$,BEGIN, VA                ;GET PHYSICAL ADDRESS FROM 16-BIT VA
004160' 000367  174064                          SWAB    EA              ;BITS<4:5>-->BITS<12:13>
004164' 006167  174060                          ROL     EA              ;NOW BITS<14:13>
004170' 006167  174054                          ROL     EA              ;NOW BITS<15:14>
004174' 042767  003776  174046                  BIC     #3776,EA        ;CLEAR THE REST..
004202' 000207                                  RTS     PC              ;RETURN...
                                        ;
                                        ;:*************************************************************
                                        ;:*
                                        ;:*     MULTIPLIES #'S IN R3 AND R4 AND RETURNS RESULT IN
                                        ;:*     TEMP.........
                                        ;:*
                                        ;:*************************************************************
                                        ;
004204' 005067  174056           MLTPLY: CLR     TEMP            ;CLEAR THE RESULT...
004210' 060467  174052           1$:     ADD     R4,TEMP         ;MULTIPLY BY ONE...
004214' 005303                           DEC     R3              ;COUNT BY ONE...
```

```
                        BNE     1$              ;NOT DONE THEN CONTINUE,
                        RTS     R5              ;RETURN...
                ;
                ;;**************************************************
                ;;*
                ;;*     LINK TABLE TO INTERRUPT SERVICE ROUTINE.
                ;;*
                ;;**************************************************
                ;
004222'         INTLNK:
004222' 004567  176056          JSR     R5,INISR        ;INPUT INTERRUPT SERVICE ROUTINE.
004226' 004567  176402          JSR     R5,OUISR        ;OUTPUT INTERRUPT SERVICE ROUTINE.
004232' 000000          CSRG1:  .WORD   0               ;CSR ADDRESS FOR DEV#1.
004234' 000000          XX11:   .WORD   0               ;END PASS FLAG FOR DEV#1.
004236' 005272'                 RBUF1                   ;RECEIVE BUFFER POINTER FOR DEV #1.
004240' 000000          XX12:   .WORD   0               ;REC/XMITR COUNTERS.
004242' 000000                  .WORD   0               ;ERROR COUNTS FOR DEV #1.
                ;
004244' 004567  176034          JSR     R5,INISR        ;INPUT INTERRUPT SERVICE ROUTINE.
004250' 004567  176360          JSR     R5,OUISR        ;OUTPUT INTERRUPT SERVICE ROUTINE.
004254' 000000          CSR22:  .WORD   0               ;CSR ADDRESS FOR DEV #2.
004256' 000000          XX21:   .WORD   0               ;END PASS FLAG FOR DEV #2.
004260' 005274'                 RBUF2                   ;RECEIVE BUFFER POINTER FOR DEV #2.
004262' 000000          XX22:   .WORD   0               ;REC/XMITR COUNTERS.
004264' 000000                  .WORD   0               ;ERROR COUNTS FOR DEV #2.
                ;
                ;;**************************************************************
                ;;*
                ;;*     BUFFERS & QUEUES.
                ;;*
                ;;**********************************************
                ;
004266' 000400  001402  002404  XBUF:   .ASCII  <000><001><002><003><004><005><006><007><010><011><012>
004274' 003406  004410  000012
004301' 000013  006414  007416          .ASCII  <013><014><015><016><017><020><021><022><023><024><025>
004306' 010420  011422  012424
004314' 013426  014430  015432          .ASCII  <026><027><030><031><032><033><034><035><036><037><040>
004322' 016434  017436  000040
004327' 000041  021442  022444          .ASCII  ^!"#$%&'()*+,-./0123456789:<^
004334' 023446  024450  025452
004342' 026454  027456  030460
004350' 031462  032464  033466
004356' 034470  036072
004362' 037075  040077  041101          .ASCII  /=>?@ABCDEFGHIJKLMNOPQRSTUVWX/
004370' 042103  043105  044107
004376' 045111  046113  047115
004404' 050117  051121  052123
004412' 053125  054127
004416' 055131  056133  057135          .ASCII  /YZ[\]^-\ABCDEFGHIJKLMNOPQR/
004424' 056055  041101  042103
004432' 043105  044107  045111
004440' 046113  047115  050117
004446' 051121
004450' 054123  053125  054127          .ASCII  /SXUVWXYZ[1]/<177><200><201><202>
004456' 055131  030533  077535
004464' 100600      202
```



```
004467'     203  102604  103606          .ASCII  <203><204><205><206><207><210><211><212><213><214><215>
004474' 104610  105612  106614
004502' 107616  110620  111622          .ASCII  <216><217><220><221><222><223><224><225><226><227><230>
004510' 112624  113626      230
004515'     231  115632  116634          .ASCII  <231><232><233><234><235><236><237><240><241><242><243>
004522' 117636  120640  121642
004530' 122644  123646  124650          .ASCII  <244><245><246><247><250><251><252><253><254><255><256>
004536' 125652  126654      256
004543'     257  130660  131662          .ASCII  <257><260><261><262><263><264><265><266><267><270><271>
004550' 132664  133666  134670
004556' 135672  136674  137676          .ASCII  <272><273><274><275><276><277><300><301><302><303><304>
004564' 140700  141702      304
004571'     305  143706  144710          .ASCII  <305><306><307><310><311><312><313><314><315><316><317>
004576' 145712  146714  147716
004604' 150720  151722  152724          .ASCII  <320><321><322><323><324><325><326><327><330><331><332>
004612' 153726  154730      332
004617'     333  156734  157736          .ASCII  <333><334><335><336><337><340><341><342><343><344><345>
004624' 160740  161742  162744
004632' 163746  164750  165752          .ASCII  <346><347><350><351><352><353><354><355><356><357><360>
004640' 166754  167756      360
004645'     361  171762  172764          .ASCII  <361><362><363><364><365><366><367><370><371><372><373>
004652' 173766  174770  175772
004660' 176774  177776      377         .ASCII  <374><375><376><377><377>
004665' 000400                          .BLKB   400                             ;
                                        .EVEN
005266' 005272'         BUFTAB: RBUF1                   ;BUFFER POINTER FOR DEV #1.
005270' 005274'                 RBUF2                   ;BUFFER POINTER FOR DEV #2.
                ;;*
                ;;*     TABLE OF RECEIVE BUFFERS.
                ;;*
005272'         RBUF1:                          ;RECEIVE BUFFERS FOR DEV #1.
005272' 005276'         RBUF11
005274'         RBUF2:                          ;RECEIVE BUFFERS FOR DEV #2.
005274' 006276'         RBUF21
                ;;*
                ;;*     RECEIVE BUFFERS FOR DEVICE #1.
                ;;*
005276' 001000  RBUF11: .BLKB   1000            ;RECEIVE BUFFER 11.
                ;;*
                ;;*     RECEIVE BUFFERS FOR DEVICE #2.
                ;;*
006276' 001000  RBUF21: .BLKB   1000            ;RECEIVE BUFFER 21.
007276'         BASE1:
                ;;*
                ;;*     QUEUES AND ITS POINTERS.
                ;;*
007276' 000010  PIRINQ: .BLKW   10              ;INPUT INTERRUPT QUEUE.
007316' 000010  PIROUTQ:.BLKW   10              ;OUTPUT INTERRUPT QUEUE.
007336' 000010  REGQ:   .BLKW   10              ;
007356' 000000  INQIN:  .WORD   0               ;INPUT QUEUE POINTER.
```

```
1074  007360' 000000          INQOUT: .WORD   0               ;INPUT QUEUE POINTER.
1075  007362' 000000          OUTQIN: .WORD   0               ;OUTPUT QUEUE POINTER.
1076  007364' 000000          OUTQOUT:.WORD   0               ;OUTPUT QUEUE POINTER.
1077  007366' 000000          REGQI:  .WORD   0               ;REGISTER QUEUES.
1078  007370' 000000          REGQO:  .WORD   0               ;REGISTER QUEUES.
1079  007372'                 MICRCD:
1080                          ;
1081                          ;;*****************************************************************
1082                          ;;*
1083                          ;;*     DEC/X11 MICRO-CODE FOR KMC11 IT WORKS AS FOLLOWS:
1084                          ;;*     1) INTERRUPT 11 PROGRAM WHEN
1085                          ;;*          A. IEI & RDI SET - INTERRUPT AT LOC. XX0 FOR REQUEST OF
1086                          ;;*     RECEIVE OR XMIT BA/CC I.
1087                          ;;*          B. IEO & RDO SET - INTERRUPT AT LOC. XX4 FOR REPORTING
1088                          ;;*     ERRORS (BY CNTL/D) OR REPORTING THE JOB DONE.
1089                          ;;*     2) WHEN RDI CLEARED BY THE 11 PROGRAM IT GOES AND
1090                          ;;*     SERVES THE REQUEST I.E. EITHER RECEIVE OR XMIT.
1091                          ;;*          A. RECEIVE OPERATION IS TO RECEIVE A BUFFER FROM
1092                          ;;*     KMC11 TO PDP11.
1093                          ;;*          B. TRANSMIT OPERATION IS TO TRANSMIT A BUFFER FROM
1094                          ;;*     PDP11 TO KMC11.
1095                          ;;*     3) IN NORMAL CASE (BOTH RECEIVE & XMIT) XMIT HAS TO BE DONE
1096                          ;;*     FIRST THEN RECEIVE.
1097                          ;;*     4) IT CONTROLS NPR RATE BY INTERNAL SOFTWARE CLOCK & NPR/BR
1098                          ;;*     RATE IS CONTROLLED BY CHARACTER COUNT.
1099                          ;;*     5) IN CASE OF XMIT OPERATION DATA CHECKING IS DONE
1100                          ;;*     INSIDE KMC11.
1101                          ;;*
1102                          ;;*****************************************************************
1103                          ;
1104                          ;
1105  007372'                 KMAAMC: MOVE    # 0,BREG                 ;CLEAR B REGISTER
1106  007374'                         MOVE    # 0,MLR          ;MAR <0:7>:=0.
1107  007376'                         MOVE    # 0,MPR          ;MAR <8:0>:=0.
1108  007400'                         MOVE    BREG,SPAD <0>   ;CLEAR SPAD  <0>.
1109  007402'                         MOVE    BREG,SPAD <1>   ;CLEAR SPAD  <1>.
1110  007404'                         MOVE    BREG,SPAD <2>   ;CLEAR SPAD  <2>.
1111  007406'                         MOVE    BREG,SPAD <3>   ;CLEAR SPAD  <3>.
1112  007410'                         MOVE    BREG,SPAD <4>   ;CLEAR SPAD  <4>.
1113  007412'                         MOVE    BREG,SPAD <5>   ;CLEAR SPAD  <5>.
1114  007414'                         MOVE    BREG,SPAD <6>   ;CLEAR SPAD  <6>.
1115  007416'                         MOVE    BREG,SPAD <7>   ;CLEAR SPAD  <7>.
1116  007420'                         MOVE    BREG,SPAD <10>  ;CLEAR SPAD  <10>.
1117  007422'                         MOVE    BREG,SPAD <17>  ;CLEAR SPAD  <17>.
1118  007424'                         MOVE    BREG,OUT1 <0>   ;CLEAR BSEL0.
1119  007426'                         MOVE    BREG,OUT1 <2>   ;CLEAR BSEL2.
1120  007430'                         MOVE    BREG,OUT1 <3>   ;CLEAR BSEL3.
1121  007432'                         MOVE    BREG,OUT1 <4>   ;CLEAR BSEL4.
1122  007434'                         MOVE    BREG,OUT1 <5>   ;CLEAR BSEL5.
1123  007436'                         MOVE    BREG,OUT1 <6>   ;CLEAR BSEL6.
1124  007440'                         MOVE    BREG,OUT1 <7>   ;CLEAR BSEL7.
1125  007442'                         MOVE    BREG,OUT1 <10>  ;
1126  007444'                         MOVE    BREG,OUT1 <11>  ;
1127  007446'                         MOVE    BREG,OUT0 <0>   ;
1128  007450'                         MOVE    BREG,OUT0 <1>   ;
1129  007452'                         MOVE    BREG,OUT0 <2>   ;
```



```
1130  007454'                         MOVE    BREG,OUT0 <3>   ;
1131  007456'                         MOVE    BREG,OUT0 <4>   ;
1132  007460'                         MOVE    BREG,OUT0 <5>   ;
1133  007462'                         MOVE    BREG,OUT0 <6>   ;
1134  007464'                         MOVE    BREG,OUT0 <7>   ;
1135  007466'                         MOVE    BREG,OUT0 <10>  ;
1136  007470'                 1$:     MOVE    # 0,MEM,MARINC          ;CLEAR MEMORY LOCATION & INC MAR
1137  007472'                         $INC    SPAD <0>                ;INCREMENT THE COUNT
1138  007474'                         $ADC    SPAD <1>                ;ADD CARRY IN TO MSB.
1139  007476'                         MOVE    SPAD <1>, BREG          ;IS IT DONE...
1140  007500'                         BB1     STRTS                   ;BR IF DONE AND START.
1141  007502'                         $BR     1$                      ;ELSE, CONTINUE
1142  007504'                 STRTS:  MOVE    # 200,BREG      ;SET RD I BIT IN B REGISTER.
1143  007506'                         MOVE    INP1 <CSR0>,SPAD <0>    ;SAVE SEL0 IN SPAD <0>.
1144  007510'                         OR      BREG,SPAD <0>,OUT1 <CSR0>       ;SET RD I IN SEL0.
1145  007512'                 MCDLP:  MOVE    INP1 <CSR2>,BREG        ;IS RDO SET???
1146  007514'                         BB7     MCDLP                   ;WAIT TILL IT CLEARS...
1147  007516'                         MOVE    INP1 <CSR0>,BREG        ;LOAD BREG WITH SEL0.
1148  007520'                         BB7     1$              ;IF RD I SET, GO CHECK IEI.
1149  007522'                         $BR     PRCSBF          ;IF CLEARED, PROCESS BUFFERS.
1150  007524'                 1$:     BB4     INTRPT          ;IF IGI IS ALSO SET THEN GO INTERRUPT
1151  007526'                         $BR     MCDLP           ;ELSE WAIT.
1152  007530'                 INTRPT:
1153  007530'                         MOVE    # 0,BREG                ;SET UP TO CLEAR CSRS.
1154  007532'                         MOVE    BREG,OUT1 <CSR4>        ;CLEAR BSEL4.
1155  007534'                         MOVE    BREG,OUT1 <CSR5>        ;CLEAR BSEL5.
1156  007536'                         MOVE    BREG,OUT1 <CSR6>        ;CLEAR BSEL6.
1157  007540'                         MOVE    BREG,OUT1 <CSR7>        ;CLEAR BSEL7
1158  007542'                         MOVE    # 200,BREG      ;SET BR REQ. WITH VECTR=XX0.
1159  007544'                         MOVE    INP1 <CSR11>,SPAD <0>   ;GET MICRO MISC. REGISTER.
1160  007546'                         OR      SPAD <0>,BREG,OUT1 <CSR11>      ;SET BR REQ. I.E. INTERRUPT AT XX0.
1161  007550'                 1$:     MOVE    INP1 <CSR11>,BREG       ;IS BR REQ BIT CLEARED.
1162  007552'                         BB7     1$              ;NO, WAIT OR SPIN ON IT.
1163  007554'                 2$:     MOVE    INP1 <CSR0>,BREG        ;IS IT SERVED???
1164  007556'                         BB7     2$              ;NO THEN WAIT...
1165  007560'                         $BR     MCDLP           ;RETURN TO LOOP.
1166                          ;
1167                          ;;*****************************************************************
1168                          ;;*
1169                          ;;*     PROCESSES THE BUFFER REQUEST.
1170                          ;;*
1171                          ;;*****************************************************************
1172                          ;
1173  007562'                 PRCSBF: MOVE    INP1 <CSR6>,SPAD <15>   ;LOAD SPAD  <15> WITH C.C LOW BYTE.
1174  007564'                         MOVE    # 77,BREG       ;LOAD MASKING BITS IN B REGISTER
1175  007566'                         MOVE    INP1 <CSR7>,SPAD <16>   ;GET HIGH BYTE OF CHARACTER COUNT.
1176  007570'                         AND     BREG,SPAD <16>,SPAD <16>        ;LOAD HIGH BYTE OF C.C. IN SPAD  <16>.
1177  007572'                 2$:     MOVE    # 7,BREG                 ;LOAD MASK IN B REG.
1178  007574'                         MOVE    INP1 <CSR0>,SPAD <0>    ;GET BSEL0.
1179  007576'                         AND     BREG,SPAD <0>,BR.SP     ;MASK OUT AND LOAD IN BREG & SPAD.
1180  007600'                         MOVE    MEM,SPAD <4>
1181  007602'                         MOVE    # 4,MEM         ;LOAD EXPECTED IN MEM.
1182  007604'                         SIFEQ MEM,SPAD <0>      XMIT    ;XMIT BA/CC I LOADED!! SERVE IT!
1183  007610'                         MOVE    # 0,MEM         ;LOAD EXPECTED IN MEM.
1184  007612'                         SIFEQ   MEM,SPAD <0>    RECVE   ;RECV BA/CC I LOADDED!! SERVE IT!
1185  007616'                         MOVE    SPAD <4>,MEM    ;RESTORE MEMORY LOCATION
```

```
1186  007620'                         MOVE    # 1,BREG                 ;SET THE TYPE OF ERROR.
1187  007622'                         MOVE    BREG,SPAD <0>           ;
1188  007624'                         CALL    SETERR          ;NONE! REPORT ERROR!!
1189  007630'                         $BR     STRTS           ;RETURN TO LOOP.
1190                                  ;
1191                                  ;!*****************************************************************
1192                                  ;!*
1193                                  ;!*     SERVE THE RECEIVE REQUEST.
1194                                  ;!*     1. DATA TRANSMIT FROM KMC11 TO PDP11. THROUGH NPR'S.
1195                                  ;!*     2. CONTROL NPR/RR & NPR RATE.
1196                                  ;!*     3. REPORT DONE OR ERROR A. NON EX MEM.
1197                                  ;!*                             B. SOFT ERROR.
1198                                  ;!*
1199                                  ;!*****************************************************************
1200                                  ;
1201  007632'                 RECVE:  MOVE    SPAD <4>,MEM    ;RESTORE MEMORY LOCATION.
1202  007634'                         MOVE    SPAD <10>,BREG  ;IS XMIT FLAG SET?
1203  007636'                         BZ      1$              ;YES, THEN SERVE THE REQUEST.
1204  007640'                         MOVE    SPAD <11>,BREG  ;IS RECEIVE ONLY FLAG SET?
1205  007642'                         BZ      1$              ;YES, THEN PROCEED TO SERVE THE REQUEST.
1206  007644'                         MOVE    # 1,BREG        ;SET THE TYPE OF ERROR...
1207  007646'                         MOVE    BREG,SPAD <0>   ;
1208  007650'                         CALL    SETERR          ;NO, REPORT ERROR!! RECV. BEFORE XMIT.
1209  007654'                         $BR     STRTS           ;WAIT FOR NEXT REQUEST.
1210  007656'                 1$:     MOVE    INP1 <CSR4>,OUT0 <6>    ;LOAD OUTBA  <0:7>
1211  007660'                         MOVE    INP1 <CSR4>,SPAD <6>    ;LOAD PARALLEL COUNT..
1212  007662'                         MOVE    INP1 <CSR5>,OUT0 <7>    ;LOAD OUTBA  <8:15>.
1213  007664'                         MOVE    INP1 <CSR5>,SPAD <7>    ;SET PARALLEL COUNT.
1214  007666'                         MOVE    INP1 <CSR7>,SPAD <0>    ;GET GET OUTBA EXTENDED BITS.
1215  007670'                         MOVE    # 300,BREG      ;GET THE MASK...
1216  007672'                         AND     BREG,SPAD <0>,BR.SP     ;
1217  007674'                         SHFBRT                  ;SET IT IN RIGHT PLACE.
1218  007676'                         SHFBRT                  ;SET IT IN RIGHT PLACE.
1219  007700'                         SHFBRT                  ;SET IT IN RIGHT PLACE.
1220  007702'                         SHFBRT                  ;SET IT IN RIGHT PLACE.
1221  007704'                         MOVE    MEM,SPAD <1>    ;SAVE MEMORY LOCATION.
1222  007706'                         MOVE    # 14,MEM        ;GET MASK
1223  007710'                         MOVE    MEM,SPAD <0>    ;IN SPAD <0>.
1224  007712'                         AND     SPAD <0>,BREG,OUT1 <11> ;SET OUT BA  <16:17>
1225  007714'                         MOVE    SPAD <1>,MEM    ;RESTORE MEMORY LOCATION.
1226  007716'                         MOVE    # 0,MLR         ;SET THE BUFFR POINTER.
1227  007720'                         MOVE    SPAD <11>,BREG  ;IS RCOLY FLAG SET?
1228  007722'                         BZ      2$              ;YES, THEN SET LOWER BUFFER ADDRESS.
1229  007724'                         MOVE    # 0,MPR         ;CLEAR THE PAGE REGISTER.
1230  007726'                         $BR     RCVLP           ;START THE TRANSMISSION.
1231  007730'                 2$:     MOVE    # 2,MPR         ;SET POINTER TO GOOD DATA BUFFER.
1232  007732'                 RCVLP:  MOVE    MEM,OUT0 <2>,MARINC     ;LOAD LOW BYTE OF DATA.
1233  007734'                         MOVE    MEM,OUT0 <3> MARINC     ;LOAD HIGH BYTE OF DATA.
1234  007736'                         MOVE    # 021,BREG      ;SET
1235  007740'                         MOVE    INP1 <10>,SPAD <0>      ;WORD XFR, NPR OUT
1236  007742'                         OR      BREG,SPAD <0>,OUT1 <10> ;       & NPR RQ..
1237  007744'                 1$:     MOVE    INP1 <10>,BREG  ;IS NPR DONE?
1238  007746'                         BB0     2$              ;NO, CHECK FOR NON EX MEM.
1239  007750'                         $BR     3$              ;YES, PREPARE FOR NEXT.
1240  007752'                 2$:     MOVE    INP1 <11>,BREG  ;IS NON EX MEM. SET?
1241  007754'                         BB0     4$              ;YES,REPORT FATAL ERROR.
```



```
1242  007756'                         $BR     1$              ;NO, WAIT FOR NPR TO CLEAR.
1243  007760'                 4$:     MOVE    # 2,BREG        ;SET THE TYPE OF ERROR...
1244  007762'                         MOVE    BREG,SPAD <0>   ;
1245  007764'                         CALL    NEXMEM          ;REPORT NON EX MEM ERROR.
1246  007770'                         $BR     STRTS           ;WAIT FOR NEXT REQUEST
1247  007772'                 3$:     $DEC    SPAD <15>       ;DECREMENT THE COUNT.
1248  007774'                         BZ      5$              ;BRANCH IF IT WAS 0.
1249  007776'                 6$:     $INC    SPAD <6>                ;UPDATE RECEIVE BUFFER
1250  010000'                         $INC    SPAD <6>                ;
1251  010002'                         $ADC    SPAD <7>                ;       POINTER.
1252  010004'                         BC      7$              ;UPDATE EXTENDED BITS IF CARRY SET.
1253  010006'                 8$:     MOVE    SPAD <6>,BREG   ;SET THE ADDRESS.
1254  010010'                         MOVE    BREG,OUT0 <6>   ;
1255  010012'                         MOVE    SPAD <7>,BREG   ;SET THE OUTBA..
1256  010014'                         MOVE    BREG,OUT0 <7>   ;
1257  010016'                         CALL    NPRATE          ;WAIT TO MAINTAIN NPR RATE.
1258  010020'                         $BR     RCVLP           ;DO THE NEXT XFR.
1259  010024'                 7$:     $DEC    SPAD <7>        ;SET BACK TO 377.
1260  010026'                         MOVE    # 4,BREG                ;LOAD LSB OF EXTEDED ADDRESS.
1261  010030'                         MOVE    INP1 <11>,SPAD <0>      ;GET EXTENDED ADDRESS BITS.
1262  010032'                         $ADD    BREG,SPAD <0>,OUT1 <11> ;INCREMENT EXTENDED BITS.
1263  010034'                         $BR     8$              ;WAIT TO MAINTAIN RATE THEN PROCEED.
1264  010036'                 5$:     $DEC    SPAD <16>       ;DECREMENT THE MSP'S.
1265  010040'                         BZ      RCVDNE          ;BRANCH IF DONE.
1266  010042'                         $BR     6$              ;ELSE, DO THE NEXT.
1267  010044'                 RCVDNE: MOVE    # 201,BREG      ;SET RD O & RCV DONE BITS IN BREG.
1268  010046'                         MOVE    INP1 <CSR2>,SPAD <0>    ;GET BSEL2.
1269  010050'                         OR      BREG,SPAD <0>,OUT1 <CSR2>       ;SET THE BITS IN BSEL2.
1270  010052'                         MOVE    INP1 <CSR2>,BREG        ;DUMMY XFR TO CHECK IEO?
1271  010054'                         BB4     1$              ;BRANCH IF IEO SET.
1272  010056'                         $BR     STRTS           ;WAIT FOR NEXT REQUEST.
1273  010060'                 1$:     MOVE    # 300,BREG      ;SET BR REQ. VCTR:=XX4
1274  010062'                         MOVE    INP1 <CSR11>,SPAD <0>   ;GET MICRO-PROCESSOR MISC. REG.
1275  010064'                         OR      BREG,SPAD <0>,OUT1 <CSR11>      ;SET THE BITS IN MISC. REG.
```

```
1276  010066'                    2$:     MOVE    INP1 <CSR11>,BREG       ;IS BR REQ CLEARED?
1277  010070'                            BB7     2$              ;NO, SPIN ON IT.
1278  010072'                            MOVE    # 277,BREG      ;SET TO CLEAR VCTR:=XX4...
1279  010074'                            MOVE    INP1 <CSR11>,SPAD <0>   ;GET UPMS REGISTER...
1280  010076'                            AND     BREG,SPAD <0>,OUT1 <CSR11>      ;CLEAR VCTR:=XX4...
1281  010100'                            MOVE    # 0,BREG                ;GET SET TO CLEAR XMIT FLAG.
1282  010102'                            MOVE    BREG,SPAD <10>  ;CLEAR XMIT FLAG.
1283  010104'                            $BR     STRTS           ;WAIT FOR NEXT REQUEST.
1284                                    ;
1285                                    ;;*****************************************************************
1286                                    ;;*
1287                                    ;;*     TRANSMIT THE BUFFER FORM PDP11 TO KMC11
1288                                    ;;*     DOES ALL THE FUNCTIONS AS RECEIVE OPERATION AND
1289                                    ;;*     ALSO CHECKS THE DATA.
1290                                    ;;*
1291                                    ;;*****************************************************************
1292  010106'                    XMIT:   MOVE    SPAD <4>,MEM    ;RESTORE MEMORY LOCATION.
1293  010110'                            MOVE    INP1 <CSR4>,OUT0 <4>    ;GET LOW BYTE OF BUFFER ADDRESS.
1294  010112'                            MOVE    INP1 <CSR4>,SPAD <6>    ;SET PARALLEL ADDRESS IN SCRATCH PAD.
1295  010114'                            MOVE    INP1 <CSR5>,OUT0 <5>    ;GET HIGH BYTE OF BUFFER ADDRESS.
1296  010116'                            MOVE    INP1 <CSR5>,SPAD <7>    ;SET PARALLEL ADDRESS IN SCRATCH PAD.
1297  010120'                            MOVE    INP1 <CSR7>,SPAD <0>    ;GET EXTENDED BITS IN BREG.
1298  010122'                            MOVE    # 300,BREG              ;GET THE MASK...
1299  010124'                            AND     BREG,SPAD <0>,BR,SP     ;.....
1300  010126'                            SHFBRT                  ;POSITION THE BITS.
1301  010130'                            SHFBRT                  ;POSITION THE BITS.
1302  010132'                            SHFBRT                  ;POSITION THE BITS.
1303  010134'                            SHFBRT                  ;POSITION THE BITS.
1304  010136'                            MOVE    MEM,SPAD <1>    ;SAVE MEMORY LOC.
1305  010140'                            MOVE    # 14,MEM                ;
```



```
1306  010142'                            MOVE    MEM,SPAD <0>    ;GET THE MASKING BITS.
1307  010144'                            AND     BREG,SPAD <0>,OUT1 <10> ;LOAD EXTENDED ADDRESS BITS OF BUFFER
1308  010146'                            MOVE    SPAD <1>,MEM    ;RESTORE MEMORY.
1309  010150'                            MOVE    # 0,MLR         ;CLEAR MEMORY LOCATION REGISTER
1310  010152'                            MOVE    # 0,MPR         ;CLEAR MEMORY PAGE REGISTER.
1311  010154'                    XMTLP:  MOVE    INP1 <CSR10>,SPAD <0>   ;GET NPR CONTROL REGISTER.
1312  010156'                            MOVE    # 001,BREG       ;SET NPR RQ. BIT IN BREG.
1313  010160'                            OR      BREG,SPAD <0>,OUT1 <CSR10>      ;SET NPR RQ BIT.
1314  010162'                    1$:     MOVE    INP1 <CSR10>,BREG       ;IS NPR DONE?
1315  010164'                            BB0     2$              ;NO, CHECK FOR NON EX. MEM
1316  010166'                            $BR     3$              ;YES, PREPARE FOR NEXT.
1317  010170'                    2$:     MOVE    INP1 <CSR11>,BREG       ;IS NON EX. MEM. SET?
1318  010172'                            BB0     4$              ;REPORT FATAL ERROR IF IT IS.
1319  010174'                            $BR     1$              ;NO, WAIT FOR NPR TO CLEAR.
1320  010176'                    4$:     MOVE    # 2,BREG        ;SET THE TYPE OF ERROR...
1321  010200'                            MOVE    BREG,SPAD <0>   ;
1322  010202'                            CALL    NEXMEM          ;REPORT NON EX. MEM. ERROR.
1323  010206'                            $BR     STRTS           ;WAIT FOR NEXT REQUEST.
1324  010210'                    3$:     MOVE    INP0 <0>,MEM MAR INC    ;LOAD THE DATA IN TO MEMORY.
1325  010212'                            MOVE    INP0 <1>,MEM MAR INC    ;LOAD THE DATA IN TO MEMORY.
1326  010214'                            $DEC    SPAD <155>              ;DECREMENT THE COUNT
1327  010216'                            BZ      5$              ;BRANCH IF IT WAS 0.
1328  010220'                    6$:     $INC    SPAD <6>                ;UPDATE THE XMIT
1329  010222'                            $INC    SPAD <6>                ;  "  "  "  "  .
1330  010224'                            $ADC    SPAD <7>                ;BUFFER POINTER.
1331  010226'                            BC      7$              ;UPDATE EXTENDED ADDRESS IF CARRY SET.
1332  010230'                    8$:     MOVE    SPAD <6>,BREG   ;SET OUTBA ADDRESS.
1333  010232'                            MOVE    BREG,OUT0 <4>   ;
1334  010234'                            MOVE    SPAD <7>,BREG   ;SET OUTBA ADDRESS.
1335  010236'                            MOVE    BREG,OUT0 <5>   ;
1336  010240'                            CALL    NPRATE          ;WAIT TO MAINTAIN THE NPR RATE.
1337  010244'                            $BR     XMTLP           ;DO THE NEXT XFR.
1338  010246'                    7$:     $DEC    SPAD <7>        ;SET IT BACK TO 377...
1339  010250'                            MOVE    # 4,BREG        ;LOAD LSB OF EXTENDED ADDRESS.
1340  010252'                            MOVE    INP1 <CSR10>,SPAD <4>   ;LOAD BREG WITH EXTENDED BITS.
1341  010254'                            $ADD    BREG,SPAD <4>,OUT1 <CSR10>      ;INCREMENT THE EXTENDED BITS.
1342  010256'                            $BR     8$              ;WAIT TO MAINTAIN RATE THEN PROCEED.
1343  010260'                    5$:     $DEC    SPAD <16>       ;DECREMENT THE MSB'S
1344  010262'                            BZ      XMTDNE          ;BRANCH IF DONE.
1345  010264'                            $BR     6$              ;ELSE DO THE NEXT.
1346  010266'                    XMTDNE: MOVE    INP1 <CSR6>,SPAD <15>   ;GET LOW BYTE OF CHAR. COUNT.
1347  010270'                            MOVE    # 77,BREG       ;LOAD THE MASK.
1348  010272'                            MOVE    INP1 <CSR7>,SPAD <16>   ;GET HIGH BYTE OF CHAR. COUNT.
1349  010274'                            AND     BREG,SPAD <16>, SPAD <16>       ;LOAD HIGH BYTE OF CHAR. COUNT.
1350  010276'                            MOVE    # 0,BREG
1351  010300'                            MOVE    BREG,SPAD <1>           ;SET POINTERS TO DATA
1352  010302'                            MOVE    BREG,SPAD <2>           ;BUFFERS.
1353  010304'                            MOVE    BREG,SPAD <3>           ;SET POINTER TO GOOD.
1354  010306'                            MOVE    # 2,BREG
1355  010310'                            MOVE    BREG,SPAD <4>           ;
1356  010312'                            MOVE    SPAD <1>,MLR            ;
1357  010314'                            MOVE    SPAD <2>,MPR            ;
1358  010316'                    CKDTLP: MOVE    MEM,SPAD <5>            ;LOAD GOOD DATA.
1359  010320'                            MOVE    SPAD <3>,MLR            ;
1360  010322'                            MOVE    SPAD <4>,MPR            ;
1361  010324'                            $IFEQ   SPAD <5>,MEM    1$      ;COMPARE DATA. GO TO 1$ IF GOOD
```

```
1362  010330'                        MOVE    # 20,BREG              ;SET THE TYPE OF ERROR...
1363  010332'                        MOVE    BREG,SPAD <0>         ;
1364  010334'                        CALL    DATERR          ;REPORT DATA ERROR.
1365  010340'                1$:     $DEC    SPAD <15>             ;DECREMENT COUNT
1366  010342'                        BZ      2$                    ;BRANCH IF LOW BYTE CIRCLED
1367  010344'                9$:     $INC    SPAD <3>              ;UPDATE GOOD DATA
1368  010346'                        $ADC    SPAD <4>              ;BUFFER POINTER
1369  010350'                        $INC    SPAD <1>        ;UPDATE DATA BUFFER POINTER
1370  010352'                        $ADC    SPAD <2>        ;AND LOAD IT IN MAR.
1371  010354'                        MOVE    SPAD <1>,MLR    ;SET THE MAR:::
1372  010356'                        MOVE    SPAD <2>,MPR    ;SET THE MAR:::
```



```
1373  010360'                        $BR     CKDTLP          ;CHECK THE NEXT CHAR.
1374  010362'                2$:     $DEC    SPAD <16>       ;DECREMENT MSP COUNT.
1375  010364'                        BZ      3$              ;BRANCH IF DONE.
1376  010366'                        $BR     9$              ;PREPARE FOR THE NEXT.
1377  010370'                3$:     MOVE    # 205,BREG      ;SET RD O, XMIT DONE BA/CC O /N BREG.
1378  010372'                        MOVE    INP1 <CSR2>,SPAD <0>  ;GET BSEL2.
1379  010374'                        OR      BREG,SPAD <0>,OUT1 <CSR2>       ;SET BITS IN BSEL2.
1380  010376'                        MOVE    INP1 <CSR2>,BREG                ;IS ISO SET?
1381  010400'                        BB4     4$              ;YES, GO INTERRUPT
1382  010402'                        $BR     STRTS           ;WAIT FOR NEXT INSTRUCTION. REQUEST.
1383  010404'                4$:     MOVE    # 300,BREG      ;
1384  010406'                        MOVE    INP1 <CSR11>,SPAD <0>   ;
1385  010410'                        OR      BREG,SPAD <0>, OUT1 <CSR11>     ;SET BR RQ,VCTR=XX4
1386  010412'                5$:     MOVE    INP1 <CSR11>,BREG       ;IS BR RQ CLEARED
1387  010414'                        BB7     5$              ;NO, SPIN ON IT.
1388  010416'                        MOVE    # 277,BREG      ;SET TO CLEAR VCTR:=XX4.
1389  010420'                        MOVE    INP1 <CSR11>,SPAD <0>   ;GET UPMS REGISTER...
1390  010422'                        AND     BREG,SPAD <0>,OUT1 <CSR11>      ;CLEAR VCTR:XX4.
1391  010424'                        $DEC    SPAD <10>       ;SET XMIT FLAG
1392  010426'                        $BR     STRTS           ;GO TO IDLE STATE.
1393                          ;
1394                          ;;**************************************************************
1395                          ;;*
1396                          ;;*     SUBROUTINES     1. NPR RATE CONTROL     2. ERROR REPORTING.
1397                          ;;*
1398                          ;;**************************************************************
1399                          ;*
1400                          ;*      1. NPRATE.  THIS ROUTINE DELAYS THE NPR TO
1401                          ;*         MAINTAIN THE RATE SET BY THE OPERATOR.
1402                          ;*
1403                          ;*         THE SMALLEST UNIT FOR RATE IS 2 MICRO-SECOND WHICH
1404                          ;*         IS SIMULATED IN SOFTWARE USING MICRO-PROCESSOR
1405                          ;*         MOVE INSTRUCTION.  DELAY COUNT IS SET AT THE TIME OF
1406                          ;*         LOADING THE MICRO-CODE.
1407                          ;*
1408  010430'                NPRATE: MOVE    BREG,SPAD <17>          ;SAVE THE RETURN ADDRESS.
1409  010432'                        MOVE    SPAD <13>,BREG          ;GET NPR RATE COUNT...
1410  010434'                        MOVE    BREG,SPAD <0>           ;
1411  010436'                        MOVE    SPAD <14>,BREG          ;COUNT MOVE BREG,SPAD <17>
1412  010440'                        MOVE    BREG,SPAD <1>           ;
1413  010442'                1$:     $DEC    SPAD <0>                        ;DECREMENT THE COUNT
1414  010444'                        BZ      2$                      ;BRANCH IF LOW BYTE IS COUNTED
1415  010446'                        $BR     1$                      ;CONTINUE
1416  010450'                2$:     $DEC    SPAD <1>                        ;COUNT HIGH BYTE BY 1.
1417  010452'                        BZ      3$                      ;RETURN IF DONE.
1418  010454'                        $BR     1$                      ;CONTINUE.
1419  010456'                3$:     $BR     SPAD <17> PAGE0         ;RETURN.
1420                          ;
1421                          ;;**************************************************************
1422                          ;;*
1423                          ;;*     THIS ROUTINE REPORTS ERROR TO PDP11 MONITOR PROGRAM...
1424                          ;;*     I.  SOFT ERROR:- (SPAD <4> BIT0:=1.)
1425                          ;;*
1426                          ;;*     II. NON EXISTENT MEMORY:- (SPAD <4> BIT1:=1.)
1427                          ;;*
1428                          ;;*     III.DATA ERROR:- (SPAD <4> BIT4:=1.)
```

```
                                ;:*
                                ;:*****************************************************************
                                ;
                                SFTERR:                                 ;ENTRY FOR SOFT ERROR.
                                NEXMEM:                                 ;ENTRY FOR NON EXISTENT MEMORY.
                                DATERR: MOVE    BREG,SPAD <17>          ;ENTRY FOR DATA ERROR.
                                        MOVE    SPAD <0>,BREG           ;FIND THE TYPE OF ERROR.
                                        BBO     SFTER1                  ;SET UP FOR SOFT ERROR.
                                        BB1     NEXME1                  ;SET UP FOR NON EX MEM.
                                        BB4     DATER1                  ;SET UP FOR DATA ERROR.
                                SFTER1: MOVE    # 212,BREG              ;SET SFTER,CNTL/0,RD 0.
                                        SBR     MERGE                   ;
                                NEXME1: MOVE    # 242,BREG              ;SET NEXMEM,CNTL/0,RD 0.
                                        SBR     MERGE                   ;
                                DATER1: MOVE    # 312,BREG              ;SET DATAERR,CNTL/0,RD 0.
                                MERGE:  MOVE    INP1 <CSR2>,SPAD <0>    ;GET BSEL2...
                                        OR      BREG,SPAD <0> OUT1 <CSR2>       ;SET THE BITS IN BSEL2.
                                        MOVE    SPAD <4>,BREG   ;LOAD ERROR DATA.
                                        MOVE    BREG,OUT1 <CSR4>        ;
                                        MOVE    SPAD <5>,BREG           ;LOAD ERROR DATA.
                                        MOVE    BREG,OUT1 <CSR5>        ;
                                        MOVE    SPAD <6>,BREG           ;LOAD ERROR DATA.
                                        MOVE    BREG,OUT1 <CSR6>        ;
                                        MOVE    SPAD <7>,BREG           ;LOAD ERROR DATA.
                                        MOVE    BREG,OUT1 <CSR7>        ;
                                        MOVE    SPAD <17>,BREG          ;LOAD ERROR DATA...
                                        MOVE    BREG,OUT1 <CSR3>        ;
                                        MOVE    INP1 <CSR2>,BREG                ;IS IE 0 SET?
                                        BB4     2$                      ;YES,GO INTERRUPT.
                                        SBR     3$                      ;NO,RETURN.
                                2$:     MOVE    # 300,BREG              ;SET BR REQ.,VCTR:=XX4.
                                        MOVE    INP1 <11>,SPAD <0>              ;GET UPROCESSOR MISC. REGISTER.
                                        OR      BREG,SPAD <0> OUT1 <11> ;SET BR REQ.,VCTR:=XX4...
                                4$:     MOVE    INP1 <11>,BREG          ;IS BR GRANTED?
                                        BB7     4$                      ;NO,SPIN ON IT.
                                        MOVE    # 277,BREG              ;SET TO CLEAR VCTR:=XX4.
                                        MOVE    INP1 <CSR11>,SPAD <0>   ;GET UPMS REGISTER...
                                        AND     BREG,SPAD <0>,OUT1 <CSR11>      ;CLEAR XX4.
                                3$:     $BR     SPAD <17> PAGE0         ;RETURN TO MAINLAND.
                                        .WORD   0                       ;MICRO-CODE TERMINATOR..
                                        .END
```



```
ACSR    000102R          222#    551*    730*
ADDR    000006R          188#    357     384     418
ADDR22= 001000           235#
ASB     000106R          226#    504*
ASTAT   000104R          224#    552*    731*
AWAS    000110R          227#    505*
BASE1   007276R          291    1064#
BEGIN   000000R          185#    287     350     370     437     438     483     507     514     515     548     560     647
                         659     660     671     680     710     722     725     728     734     738     754     819     944
BIT0  = 000001           235#    417     466     677     753
BIT1  = 000002           235#    413     469     667     718
BIT10 = 002000           235#    798     799     872     931
BIT11 = 004000           235#
BIT12 = 010000           235#    345
BIT13 = 020000           235#    347     354     799
BIT14 = 040000           235#    314     320     427
BIT15 = 100000           235#    432
BIT2  = 000004           235#    417     740     743     753
BIT3  = 000010           235#    413     720     748
BIT4  = 000020           235#    669     751     752
BIT5  = 000040           235#    726
BIT6  = 000100           235#    723
BIT7  = 000200           235#    669     679     753
BIT8  = 000400           235#    930     931
BIT9  = 001000           235#    928     930     931
BRCR  = 000011             1#
BREAK$= 104407           235#    437     438     514     515
BR1     000012R          190#    404     408
BR2     000013R          191#
BTOD$ = 104421           235#
BUFTAB  005266R          474    1045#
CDATA$= 104412           235#
CKDTLP  010316R         1358#   1374
CONFIG  000056R          210#
CRMOFW  003506R          819     822#
CSRA    000100R          220#    501*    550*    729*
CSRC1   004232R          528     974#
CSR0  = 000000             1#   1144    1145    1148    1164    1179
CSR1  = 000001             1#
CSR10 = 000010             1#   1312    1314    1315    1341    1342
CSR11 = 000011             1#   1160    1161    1162    1275    1276    1277    1280    1281    1318    1385    1386    1387
                        1390    1391    1466    1467
CSR2  = 000002             1#   1146    1269    1270    1271    1379    1380    1381    1445    1446    1457
CSR22   004254R          533     982#
CSR3  = 000003             1#   1456
CSR4  = 000004             1#   1155    1211    1212    1294    1295    1448
CSR5  = 000005             1#   1156    1213    1214    1296    1297    1450
CSR6  = 000006             1#   1157    1174    1347    1452
CSR7  = 000007             1#   1158    1176    1215    1298    1349    1454
DATCK$= 104411           235#
DATERR  010460R         1365    1434#
DATER$= 104404           235#    507
DATER1  010502R         1439    1443#
DATIN = 000001             1#
DATI0 = 000000             1#
DATI1 = 000001             1#
```

```
DATOUT= 000021            1#
DATOO = 000002            1#
DATO1 = 000003            1#
DEV     002004R   546*    547*    564#    566
DLV1    000224R   244#    462*    520*    553*    618
DLV2    000216R   245#    463*    517*    554*    619
DROP    000316R   293     296#    294     351     374
DROP1   002006R   548     565#
DTER1   003162R   725     759#
DVID1   000014R   192#    282     284
EA      000250R   254#    665     675     945*    946*    947*    948*
EABITS  004152R   663     673     943#
ENDITS= 104413    235#    483
ENDS    = 104410  235#    287
ERRTYP  000106R   225#    558*    732*
ESAV1   002246R   555*    611#    620     624
ESAV2   002250R   556*    612#    621     625
ESAV3   002252R   557*    613#    622     626
ESAV4   002254R   614#    627
ETABLE  002256R   560     618#
EXITS   = 104400  235#    660     671     680     738     754
FIRST   000234R   248#    288*    425     481*
FLAG    000270R   265#    281*    295     482*    904*    908     909     922*
FLAGB   000232R   247#    387     388     464     467     749
FRCPFW= 000002            1#
FTABLE  002272R   624#
GETPAS= 104415    235#    944
GWBUFS= 104414    235#
HRDCNT  000044R   202#
HRDERS= 104405    235#    560     734
HRDPAS  000050R   207#
IBAD0 = 000004            1#
IBAD1 = 000005            1#
IBA16 = 000004            1#
IBA17 = 000010            1#
ICONT   000036R   202#
ICOUNT  000040R   203#
ICR     = 000012  1#
IDNUM   000122R   222#
ILINT   002554R   659     681#
INISR   002304R   640#    972     980
INIT    000030R   199#
INNPR = 000000            1#
INQIN   007356R   389*    640*    641*    642     644*    1073#
INQOUT  007360R   390*    649     650*    651     653*    1074#
INTLNK  004222P   386     472     971#
INTR    000120R   231#
INTRPT  007530R   1151    1152#
KMAAMC  007372R   794     868     1105#
LDATI = 000010            1#
LDATO = 000010            1#
LOOP    000326R   289#    485     535
LULOOP= 000040            1#
MAINT = 000017            1#
MAP22S= 104416    235#
MARHLD= 000000            1#
```



```
MARINC= 014000            1#    1137    1233    1234    1325    1326
MASK    000236R   249#    368     461*    466*    469*    470     523     867*    877*    898*    920*
MCDLP   007512R   1145#   1147    1152    1166
MERGE   010504R   1441    1443    1444#
MES1    003166R   755     761#
MES2    002560R   681     683#
MES3    003512R   922     924#
MES4    003303R   757     775#
MES5    003322R   759     778#
MICPC = 000474    1105#   1189    1209    1246    1258    1323    1337    1365
MICRCD  007372R   1079#   1105
MLRLD = 010000            1#    1107    1227    1310    1357    1360    1372
MLTPLV  004204R   334     341     959#
MODNAM  000000R   186#
MODSP   000224R   200     233#
MPRLD   = 004000          1#    1108    1230    1232    1311    1358    1361    1373
MSGNS = 104403    235#    350     370     548     659     722     725     728     819
MSGS$ = 104402    235#
MSGS    = 104401  235#
NEXMEM  010460R   1246    1323    1433#
NEXME1  010476R   1438    1441#
NPRATE  010430R   1258    1337    1408#
NPRBYT= 000200            1#
NPRC    = 000010          1#
NPRTE   000264R   263#    318*    344*
NULL    = 000000  235#    734
NXMEM = 000001            1#
NXMMRY  003152R   728     755#
OBAD0 = 000006            1#
OBAD1 = 000007            1#
OBA16 = 000004            1#
OBA17 = 000010            1#
OCR   = 000011            1#
OPEN    = 000000  187     193     194     195     196     213     214     215     216     217     218     219     220
                  222     224     226     227     229     230     231     235#

OTNPR = 000020            1#
OTOAS = 104420    235#
OUISR   002634R   703#    973     981
OUTQIN  007362R   392*    703*    704*    705     707*    1075#
OUTQOU  007364R   391*    712     713*    714     716*    1076#
PA      000246R   253#    664     674
PAGE0 = 000000            1#    1420    1468
PAGE1 = 001000            1#
PAGE2 = 002000            1#
PAGE3 = 003000            1#
PASCNT  000034R   201#
PCLK    = 000020          1#
PIRINQ  007276R   389     390     642     644     651     653     1070#
PIROUT  007316R   391     392     705     707     714     716     1071#
PIRQS = 000004    235#    647     710
POPSP = 005726    235#
POPSP2= 022626    235#
PRCSBF  007562R   1150    1173#
PRTY    = 000000  235#
PRTY0 = 000000    235#
PRTY1 = 000040    235#
```

```
PRTV2 = 000100            235#
PRTV3 = 000140            235#
PRTV4 = 000200            235#
PRTV5 = 000240            190     191     235#
PRTV6 = 000300            235#
PRTV7 = 000340            235#
PS    = 177776            235#
PSW   = 177776            235#
PUSH  = 005746            235#
PUSH2 = 024646            235#
RABO  = 000200              1#
RACT  = 000100              1#
RANDS = 104417            235#
RANNUM  000054R           209#
RBCC  = 000001              1#
RBR   = 000200              1#
RBUF1   005272R           976    1045    1051#
RBUF11  005276R           989    1052    1059#
RBUF2   005274R           994    1046    1054#
RBUF21  006276R          1055    1063#
RCSLV   000262R           260#    308#    352*    415     443     806     887
RCVDNE  010044R          1266    1267#
RCVLP   007732R          1231    1232#   1259
RECV    002426R           656     661#
RECVE   007632R          1185    1201#
REGQ    007336R           393     397     1072#
REGQI   007366R           393*    1077#
REGQO   007370R           394*    1078#
RECOM = 000002              1#
RESTRT  000322R           228     285     288#
RES1    000056R           211#
RES2    000060R           212#
ROMCLK  004126R           811     841     843     849     892     894     910     912     914     916     928#
RQNPR = 000001              1#
RRDY  = 000020              1#
RSIZE   000240R           250#    316*    337*    498     666
RSTRT   000112R           226#
RTMULV  000260R           258#    318     340
SAR0    000252R           255#    435*    439
SAR1    000254R           256#    436*    440
SBADR   000102R           221#    502*
SCAN    001334R           422     461#
SELECT  000230R           246#    284#    293     358     395     419     473     524*
SETUP1  000744R           296     362     384#
SETUP2  001166R           398     418#
SFTERR  010460R          1189    1209    1432#
SFTER1  010472R          1437    1439#
SFT1    003156R           722     757#
SOFCNT  000042R           204#
SOFERS= 104406            235#
SOFPAS  000046R           206#
SOFT1   002016R           350     570#
SOFT11  002110R           570     589#
SOFT2   002022R           370     572#
SOFT21  002171R           572     599#
SPOINT  000032R           200#
```



```
SPSIZ = 000040              1#    233
SR1     000016R           193#    312     314     320     322     323     345     347     354
SR2     000020R           194#
SR3     000022R           195#
SR4     000024R           196#
START   000272R           199     281#
STAT    000026R           198#
STRTS   007504R          1141    1142#   1190    1210    1247    1273    1284    1324    1383    1393
SVR0    000062R           213#
SVR1    000064R           214#
SVR2    000066R           215#
SVR3    000070R           216#
SVR4    000072R           217#
SVR5    000074R           218#
SVR6    000076R           219#
SW1   = 000015              1#
SW2   = 000016              1#
SYSCNT  000052R           208#
SZMULV  000256R           257#    316     317     333
TEMP    000266R           264#    335     337     338     342     344     359*    371*    372     498*    510*    959*    960*
TERM  = 000000            252#
TRPDFD= 000022            235#
VA      000244R           252#    662*    672*    944
VECTOR  000010R           189#    385
VEC4  = 000100              1#
VERIFY  003624R           367     866#
WASADR  000104R           223#    503*
WCRAM   003356R           365     793#
WDFR    000116R           230#
WDTO    000114R           229#
WMEMRY  003552R           366     840#
XABO  = 000200              1#
XACT  = 000100              1#
XBUF    004266R           497     672     845     903     994#
XDROP1  002026R           565     578#
XDROP2  002050R           567     582#
XEOM  = 000002              1#
XERR    001666R           527     532     545#
XFLAG   000005R           187#
XMIT    010106R          1183    1292#
XMITR   002502R           658     672#
XMOLV   000263R           261#    309*    356*    411     493     744
XMTDNE  010266R          1345    1346#
XMTLP   010154R          1311#   1338
XRDY  = 000020              1#
XSFU  = 000040              1#
XSIZE   000242R           251#    317*    338*    676     905
XSOM  = 000001              1#
XX11    004234R           310*    387*    464     975#
XX12    004240R           377#
XX21    004256R           311*    388*    467     983#
XX22    004262R           985#
$$INIT= 007372R          1105#   1141    1142    1147    1149    1150    1151    1153    1163    1165    1166    1183    1185
                         1189    1190    1204    1206    1209    1210    1229    1231    1239    1240    1242    1243    1246
                         1247    1249    1253    1258    1259    1264    1266    1267    1272    1273    1278    1284    1316
                         1317    1319    1320    1323    1324    1328    1332    1337    1338    1343    1345    1346    1362
```

```
                1365    1367    1374    1376    1377    1382    1383    1388    1393    1415    1416    1418    1419
                1437    1438    1439    1441    1443    1458    1459    1464
$$$DER= 000001  1106#    1107#   1108#   1109#   1110#   1111#   1112#   1113#   1114#   1115#   1116#   1117#   1118#
                1119#   1120#   1121#   1122#   1123#   1124#   1125#   1126#   1127#   1128#   1129#   1130#   1131#
                1132#   1133#   1134#   1135#   1136#   1137#   1140#   1143#   1144#   1145#   1146#   1148#   1154#
                1155#   1156#   1157#   1158#   1159#   1160#   1161#   1162#   1164#   1174#   1175#   1176#   1177#
                1178#   1179#   1180#   1181#   1182#   1183#   1194#   1195#   1196#   1187#   1188#   1189#   1202#
                1203#   1205#   1207#   1208#   1209#   1211#   1212#   1213#   1214#   1215#   1216#   1217#   1232#
                1223#   1224#   1225#   1226#   1227#   1228#   1230#   1232#   1233#   1234#   1235#   1236#   1237#
                1238#   1241#   1244#   1245#   1246#   1254#   1255#   1256#   1257#   1258#   1261#   1262#   1263#
                1268#   1269#   1270#   1271#   1274#   1275#   1276#   1277#   1279#   1280#   1281#   1282#   1283#
                1293#   1294#   1295#   1296#   1297#   1298#   1299#   1300#   1305#   1306#   1307#   1308#   1306#
                1310#   1311#   1312#   1313#   1314#   1315#   1318#   1321#   1322#   1323#   1325#   1326#   1333#
                1334#   1335#   1336#   1337#   1340#   1341#   1342#   1347#   1348#   1349#   1350#   1351#   1352#
                1353#   1354#   1355#   1356#   1357#   1359#   1359#   1360#   1361#   1362#   1363#   1364#   1365#
                1372#   1373#   1378#   1379#   1380#   1381#   1384#   1385#   1386#   1387#   1389#   1390#   1391#
                1409#   1410#   1411#   1412#   1413#   1435#   1436#   1440#   1442#   1443#   1445#   1446#   1447#
                1448#   1449#   1450#   1451#   1452#   1453#   1454#   1455#   1456#   1457#   1460#   1461#   1462#
$$$SER= 000001  1104#    1107#   1108#   1109#   1110#   1111#   1112#   1113#   1114#   1115#   1116#   1117#   1119#
                1106#   1107#   1108#   1109#   1110#   1111#   1112#   1113#   1114#   1115#   1116#   1117#   1119#
                1119#   1120#   1121#   1122#   1123#   1124#   1125#   1126#   1127#   1128#   1129#   1130#   1131#
                1132#   1133#   1134#   1135#   1136#   1137#   1140#   1143#   1144#   1146#   1146#   1154#   1155#
                1156#   1157#   1158#   1159#   1160#   1162#   1164#   1174#   1175#   1176#   1178#   1179#   1181#
                1182#   1184#   1186#   1187#   1188#   1189#   1202#   1203#   1205#   1207#   1208#   1209#   1211#
                1212#   1213#   1214#   1215#   1216#   1222#   1223#   1224#   1226#   1227#   1228#   1230#   1232#
                1233#   1234#   1235#   1236#   1238#   1241#   1244#   1245#   1246#   1254#   1255#   1256#   1257#
                1258#   1261#   1268#   1269#   1270#   1271#   1274#   1275#   1277#   1279#   1280#   1282#   1283#
                1293#   1294#   1295#   1296#   1297#   1298#   1299#   1305#   1306#   1307#   1309#   1310#   1311#
                1312#   1313#   1315#   1318#   1321#   1322#   1323#   1325#   1326#   1333#   1334#   1335#   1336#
                1337#   1340#   1341#   1347#   1348#   1349#   1351#   1352#   1353#   1354#   1355#   1356#   1357#
                1358#   1359#   1360#   1361#   1363#   1364#   1365#   1372#   1373#   1378#   1379#   1381#   1384#
                1385#   1387#   1389#   1390#   1409#   1410#   1411#   1414#   1413#   1435#   1436#   1440#   1442#
                1444#   1445#   1447#   1448#   1449#   1450#   1451#   1452#   1453#   1454#   1455#   1456#   1457#
                1460#   1461#   1463#   1466#   1466#
.       = 010566R  518     521     783#    830#    1041#   1043#   1059#   1063#   1070#   1071#   1072#   1105#
.ADC    = 000100      1#    1139    1252    1331    1369    1371
.ADD    = 000000      1#    1263    1342
.ADDWC= 000020        1#
.AND    = 000260      1#    1177    1180    1217    1225    1281    1300    1308    1350    1391    1467
.BBO    = 002000      1#    1239    1242    1316    1319    1437
.BB1    = 002400      1#    1141    1438
.BB4    = 003000      1#    1151    1272    1382    1439    1458
.BB7    = 003400      1#    1147    1149    1163    1165    1278    1388    1464
.BC     = 001000      1#    1253    1332
.BR     = 000400      1#    1142    1150    1152    1166    1189    1190    1209    1210    1231    1240    1243    1246
                1247    1258    1259    1264    1267    1273    1284    1317    1320    1323    1324    1337    1338
                1343    1346    1365    1374    1377    1383    1393    1416    1419    1420    1441    1443    1459
                1468
.BSBRG= 160000      1#    1420    1468
.BSIMM= 100000      1#    1141    1142    1147    1149    1150    1151    1152    1163    1165    1166    1183    1185
                1189    1190    1204    1206    1209    1210    1229    1231    1239    1240    1242    1243    1246
                1247    1249    1253    1258    1259    1264    1266    1267    1272    1273    1278    1234    1316
                1317    1319    1320    1323    1324    1328    1332    1337    1338    1343    1345    1346    1362
                1365    1367    1374    1376    1377    1382    1383    1388    1393    1415    1416    1418    1419
                1437    1438    1439    1441    1443    1458    1459    1464
.BSMEM= 140000      1#
```



```
.BZ     = 001400      1#    1183    1185    1204    1206    1229    1249    1266    1328    1345    1362    1367    1376
                1415    1418
.C0     = 000400      1#
.DBR    = 000400      1#    1106    1140    1143    1146    1148    1154    1159    1161    1162    1164    1175    1178
                1187    1189    1203    1205    1207    1209    1216    1225    1228    1235    1238    1241    1244
                1246    1254    1256    1258    1261    1268    1271    1274    1277    1279    1282    1299    1313
                1315    1318    1321    1323    1333    1335    1337    1340    1348    1351    1355    1363    1365
                1378    1381    1384    1387    1389    1410    1412    1436    1440    1442    1444    1447    1449
                1451    1453    1455    1457    1460    1463    1465
.DBRSH= 001400      1#    1218    1219    1220    1221    1301    1302    1303    1304
.DBRSP= 003400      1#    1180    1217    1300
.DD     = 003000  1145#    1161#   1177#   1180#   1183#   1185#   1217#   1225#   1237#   1263#   1270#   1276#   1281#
                1300#   1308#   1314#   1342#   1350#   1362#   1380#   1386#   1391#   1446#   1462#   1467#
.DEC    = 000160      1#    1248    1260    1265    1327    1339    1344    1366    1375    1392    1414    1417
.DMEM   = 002400      1#    1137    1182    1183    1184    1185    1186    1202    1223    1226    1293    1306    1309
                1325    1326    1362
.DNOP   = 000000      1#    1183    1185    1362
.DOUT0= 002000      1#    1128    1129    1130    1131    1132    1133    1134    1135    1136    1211    1213    1233
                1234    1255    1257    1294    1296    1334    1336
.DOUT1= 001000      1#    1119    1120    1121    1122    1123    1124    1125    1126    1127    1145    1155    1156
                1157    1158    1161    1225    1237    1263    1270    1276    1281    1308    1314    1342    1380
                1386    1391    1446    1448    1450    1452    1454    1456    1462    1467
.DSP    = 003000      1#    1109    1110    1111    1112    1113    1114    1115    1116    1117    1119    1138    1139
                1144    1145    1160    1174    1176    1177    1179    1180    1181    1188    1208    1212    1214
                1215    1217    1222    1224    1236    1237    1245    1248    1250    1251    1252    1260    1262
                1263    1265    1269    1270    1275    1276    1280    1281    1283    1295    1297    1298    1300
                1305    1307    1308    1312    1314    1322    1327    1329    1330    1331    1339    1341    1342
                1344    1347    1349    1350    1352    1353    1354    1356    1359    1364    1366    1368    1369
                1370    1371    1375    1379    1380    1385    1386    1390    1391    1392    1409    1411    1413
                1414    1417    1435    1445    1446    1461    1462    1466    1467
.DO     = 000400      1#
.F0     = 000020      1#    1144    1146    1148    1160    1162    1164    1174    1176    1179    1211    1212    1213
                1214    1215    1236    1238    1241    1262    1269    1271    1275    1277    1280    1294    1295
                1296    1297    1298    1312    1315    1318    1325    1326    1341    1347    1349    1379    1381
                1385    1387    1390    1445    1457    1461    1463    1466
.INC    = 000060      1#    1138    1250    1251    1329    1330    1368    1370
.LORN   = 000240      1#
.MINUS= 000360        1#
.MO     = 004000      1#
.OR     = 000300      1#    1145    1161    1237    1270    1276    1314    1380    1386    1446    1462
.PLUS   = 000000      1#
.SBR    = 060000      1#    1109    1110    1111    1112    1113    1114    1115    1116    1117    1118    1119    1120
                1121    1122    1123    1124    1125    1126    1127    1128    1129    1130    1131    1132    1133
                1134    1135    1136    1138    1139    1140    1145    1155    1156    1157    1158    1161    1177
                1180    1186    1188    1202    1203    1205    1208    1217    1218    1219    1220    1221    1225
                1226    1228    1237    1245    1248    1250    1251    1252    1254    1255    1256    1257    1260
                1263    1265    1270    1276    1281    1283    1293    1300    1301    1302    1303    1304    1308
                1309    1314    1322    1327    1329    1330    1331    1333    1334    1335    1336    1339    1342
                1344    1350    1352    1353    1354    1356    1357    1358    1360    1361    1364    1366    1368
                1369    1370    1371    1372    1373    1375    1379    1380    1386    1391    1392    1409    1410    1411
                1412    1413    1414    1417    1435    1436    1446    1447    1448    1449    1450    1451    1452
                1453    1454    1455    1456    1462    1467
.SELA = 000200      1#    1140    1186    1202    1203    1205    1226    1228    1254    1256    1293    1309    1333
                1335    1357    1358    1360    1361    1372    1373    1410    1412    1420    1436    1447    1449
                1451    1453    1455    1468
.SELB = 000220      1#    1109    1110    1111    1112    1113    1114    1115    1116    1117    1118    1119    1120
```

```
                        1121    1122    1123    1124    1125    1126    1127    1128    1129    1130    1131    1132    1133
                        1134    1135    1136    1155    1156    1157    1158    1181    1188    1208    1218    1219    1220
                        1221    1222    1224    1233    1234    1245    1255    1257    1283    1301    1302    1303    1304
                        1305    1307    1322    1334    1336    1352    1353    1354    1356    1359    1364    1409    1411
                        1413    1435    1448    1450    1452    1454    1456
.SIMM = 000000             1#   1106    1107    1108    1137    1143    1154    1159    1175    1178    1182    1184    1187
                        1189    1207    1209    1216    1223    1227    1230    1232    1235    1244    1246    1258    1261
                        1268    1274    1279    1282    1299    1306    1310    1311    1313    1321    1323    1337    1340
                        1348    1351    1355    1363    1365    1378    1384    1389    1440    1442    1444    1460    1465
.SIN0 = 020000             1#   1325    1326
.SIN1 = 120000             1#   1144    1146    1148    1160    1162    1164    1174    1176    1179    1211    1212    1213
                        1214    1215    1236    1238    1241    1262    1269    1271    1275    1277    1280    1294    1295
                        1296    1297    1298    1312    1315    1318    1341    1347    1349    1379    1381    1385    1387
                        1390    1445    1457    1461    1463    1466
.SMEM = 040000             1#   1181    1183    1185    1222    1224    1233    1234    1305    1307    1359    1362
.SUBWC= 000040             1#
.SUB1C= 000340             1#
.SUB2C= 000360             1#   1183    1185    1362
.SO   = 020000             1#
.XOR  = 000320             1#
.$.   = 160617             1#
                        1106#   1107#   1108#   1109#   1110#   1111#   1112#   1113#   1114#   1115#   1116#   1117#   1118#
                        1119#   1120#   1121#   1122#   1123#   1124#   1125#   1126#   1127#   1128#   1129#   1130#   1131#
                        1132#   1133#   1134#   1135#   1136#   1137#   1138#   1139#   1140#   1141#   1142#   1143#   1144#
                        1145#   1146#   1147#   1148#   1149#   1150#   1151#   1152#   1154#   1155#   1156#   1157#   1158#
                        1159#   1160#   1161#   1162#   1163#   1164#   1165#   1166#   1174#   1175#   1176#   1177#   1178#
                        1179#   1180#   1181#   1182#   1183#   1184#   1185#   1186#   1187#   1188#   1189#   1190#   1202#
                        1203#   1204#   1205#   1206#   1207#   1208#   1209#   1210#   1211#   1212#   1213#   1214#   1215#
                        1216#   1217#   1222#   1223#   1224#   1225#   1226#   1227#   1228#   1229#   1230#   1231#   1232#
                        1233#   1234#   1235#   1236#   1237#   1238#   1239#   1240#   1241#   1242#   1243#   1244#   1245#
                        1246#   1247#   1248#   1249#   1250#   1251#   1252#   1253#   1254#   1255#   1256#   1257#   1258#
                        1259#   1260#   1261#   1262#   1263#   1264#   1265#   1266#   1267#   1268#   1269#   1270#   1271#
                        1272#   1273#   1274#   1275#   1276#   1277#   1278#   1279#   1280#   1281#   1282#   1283#   1284#
                        1293#   1294#   1295#   1296#   1297#   1298#   1299#   1300#   1305#   1306#   1307#   1308#   1309#
                        1310#   1311#   1312#   1313#   1314#   1315#   1316#   1317#   1318#   1319#   1320#   1321#   1322#
                        1323#   1324#   1325#   1326#   1327#   1328#   1329#   1330#   1331#   1332#   1333#   1334#   1335#
                        1336#   1337#   1338#   1339#   1340#   1341#   1342#   1343#   1344#   1345#   1346#   1347#   1348#
                        1349#   1350#   1351#   1352#   1353#   1354#   1355#   1356#   1357#   1358#   1359#   1360#   1361#
                        1362#   1363#   1364#   1365#   1366#   1367#   1368#   1369#   1370#   1371#   1372#   1373#   1374#
                        1375#   1376#   1377#   1378#   1379#   1380#   1381#   1382#   1383#   1384#   1385#   1386#   1387#
                        1388#   1389#   1390#   1391#   1392#   1393#   1409#   1410#   1411#   1412#   1413#   1414#   1415#
                        1416#   1417#   1418#   1419#   1420#   1435#   1436#   1437#   1438#   1439#   1440#   1441#   1442#
                        1443#   1444#   1445#   1446#   1447#   1448#   1449#   1450#   1451#   1452#   1453#   1454#   1455#
                        1456#   1457#   1458#   1459#   1460#   1461#   1462#   1463#   1464#   1465#   1466#   1467#   1468#
.2A   = 000120             1#
.2AWC = 000140             1#

. ABS.  000000      000
        010566      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XKMCB0,XKMCB0/SOL/CRF:SYM=DDXCOM,XKMCB0
 RUN-TIME: 20 22 1 SECONDS
```



```
 RUN-TIME RATIO: 174/44=3.9
 CORE USED:  9K  (17 PAGES)
```

DIAGNOSTIC ENGINEERING

digital **DECO☐ DEPO☒ SUBMISSION☐**

FOR RELEASE ENG. USE ☐ NEW ☐ CHANGE ☐ DELETE

PRODUCT IDENTIFICATION

MD	LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE (DD MMM YY)	STATUS	DISTRIBUTION	1ST COPY-RIGHT YEAR	LAST COPY-RIGHT YEAR
	ZZ	CXKMC	B	1	Q1	19 APR 79	☐ OBSOLETE	X G ☐ R	1976	1979

TITLE: CXKMCB1 KMC-11 MODULE

AUTHOR: D. BUTENHOF | MAINTAINING GROUP: DEC/X SUPT GRP | MAINTAINER: D. BUTENHOF | SUBMITTING ENGINEER: D. BUTENHOF

PRODUCT COMPONENTS

CK	DESCRIPTION	PRODUCT NO.	REV	CK	DESCRIPTION	PRODUCT NO.	REV
	DOCUMENT				INDEX		
	LISTING				SOURCE MEDIA		
	OBJECT MEDIA				TEST MEDIA		
X	DECO	AF-E95ØB-M1					

PRODUCTS OBSOLETED (other than previous version)

	LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV
MD				M D				M D			

PRODUCT CHARACTERISTICS

PROCESSORS PRODUCT OPERATES WITH (Enter all applicable 2-digit codes representing the Processor the product operates with. See separate instructions.)

OPERATIONAL CODES (Enter all applicable 2-digit codes that describe the product. See separate instructions.)

ACT/APT/XXDP INFORMATION FIELD	EXT	ACT SEQ NUMBER	ACT/XXDP COMPATIBLE?	APT COMPATIBLE?	1ST PASS RUN TIME	SUBSEQUENT PASS RUN TIME
			X Y ☐ N	X Y ☐ N	SECONDS	SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED:

DEVICE AFFECTED: DEC/X11 | MULTIMEDIA AFFECTED? ☐ YES XX NO

KIT NUMBERS							
	ZJ13Ø-RB						
	ZJ129-RZ,	FR					

PROBLEM:

ERRORS WHEN BUFFERS OVERLAP 32K BOUNDARY, DUE TO MICROCODE BUGS.

SOLUTION:

PATCH THE FOLLOWING MODULE LOCATIONS

DEPO PATCH AREA

CHANGE LOC	FROM	TO	CHANGE LOC	FROM	TO
1ØØ24	63167	632Ø7			
1Ø246	63167	632Ø7			
1Ø254	61Ø14	61Ø1Ø			

SUBMITTING ENGINEER / MANUFACTURING ENGINEER	SUPPORT ENGINEER	CHARGE DECO/DEPO TO DISCRETE PROJECT NUMBER
A. C. Casella		
DATE: 19 APR 79 / DATE: 25-APR-79	DATE:	898Q546Q
MAINTAINER / FIELD SERVICE	WAIVERING MANAGER	COORDINATION NO. MC 3087
DATE: 25 Apr. 79	DATE:	