```
                .REM _



                                  IDENTIFICATION
                                  --------------
                   PRODUCT CODE:    AC-E691J-MC

                   PRODUCT NAME:    CXKWBJ0 DEC/X11 KW11-P MODULE

                   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)  1973,1978 DIGITAL EQUIPMENT CORPORATION
```

```
1.      ABSTRACT
        --------

        THE KWB IS AN IOMOD THAT EXERCISES THE KW11-P PROGRAMMABLE LINE
        CLOCK.  IT EXERCISES THE KW11-P AT 100KHZ, 10KHZ, AND LINE
        FREQUENCY RATES. IT DOES NOT CHECK THE EXTERNAL RATE FEATURE.
        WHENEVER LINE FREQUENCY IS SELECTED, IT IS RUN IN THE REPEAT
        INTERRUPT MODE; OTHER FREQUENCIES ARE ALWAYS RUN IN THE NON-
        REPEAT INTERRUPT MODE.
        IT ALSO ENABLES THE MONITOR TO USE THE CLOCK TO INDICATE ELAPSED TIME.

2.      REQUIREMENTS
        ------------

        HARDWARE:  ONE KW11-P CONTROL UNIT
        SOFTWARE:  CAN BE USED AS A SYSTEM CLOCK

        STORAGE:: KWB REQUIRES:
                        1. DECIMAL WORDS: 549
                        2. OCTAL WORDS: 1045
                        3. OCTAL BYTES: 2112


3.      PASS DEFINITION:
        ----------------

        ONE PASS OF THE KWB MODULE CONSISTS OF GENERATING INTERRUPTS
        FOR ONE SECOND AT EACH OF THE THREE CLOCK RATES UNTIL 60.
        SECONDS HAVE ELAPSED.

4.      EXECUTION TIME:
        ---------------

        ONE PASS OF THE KWB MODULE RUNNING ALONE ON A PDP11/05
        PROCESSOR TAKES APPROXIMATELY ONE MINUTE

5.      CONFIGURATION REQUIREMENTS:
        ---------------------------

        DEFAULT PARAMETERS:

        DEVADR: 172540, VECTOR: 104, BR1: 6, DEVCNT: 1, SR1:0

        REQUIRED PARAMETERS:

        NONE
```

```
6.      DEVICE/OPTION SET-UP:
        ---------------------

        NONE

7.      MODULE OPERATION
        ----------------

        TEST SEQUENCE:

        A.  DO 'SYSTEM CLOCK' INITIALIZATION
        B.  SETUP FOR ANY MESSAGE PRINTOUTS
        C.  SETUP CLOCK ADDRESS AND VECTOR
        D.  SETUP FOR THE PROPER LINE FREQUENCY
        E.  SETUP FOR THE NEXT CLOCK FREQUENCY
        F.  START THE CLOCK
        G.  AT THE END OF EACH SECOND, CHANGE CLOCK FREQUENCY
        H.  IF 1 MINUTE ELAPSED, CONTINUE. ELSE GO TO D
        I.  IF THIS IS NOT THE SYSTEM CLOCK, GO TO J
        J.  IF SYSTEM CLOCK HAS NOT MADE A PASS FOR 15. MINUTES,
                GO BACK TO MONITOR AND CHECK FOR HUNG MODULES.
        K.  DO AN END OF PASS
        L.  RESTART:  IF TIME TO PRINT A TIME MESSAGE, DO IT
        M.  GO TO D
```

```
8.      OPERATION OPTIONS:
        ------------------

        VALID SR1 VALUES:

                SR1     HERTZ   TIME MSG EVERY X MINUTES

                0       60              NEVER
                1       50              NEVER
                2       60              5
                3       50              5
                4       60              15              ; ALL TIMES ARE APPROXIMATE
                5       50              15
                6       60              60
                7       50              60

                10      50/60           RUN CLOCK AT LINE FREQ. ONLY
                                        (MUST ALSO SET BIT0=1 FOR 50 HZ)
                20      10,000          RUN CLOCK AT 10 KHZ ONLY
                30      100,000         RUN CLOCK AT 100 KHZ ONLY

        NOTE:   SR1 VALUES 0 THRU 7 MAY BE USED WITH SR1 VALUES 10, 20, AND 30
                TO OBTAIN MESSAGE PRINTOUTS AT A FIXED FREQUENCY.
                        FOR EXAMPLE:

                                SR1=11  RUN CLOCK AT LINE FREQ. ONLY,
                                        LINE FREQ. IS 50 HZ

                                SR1=36  RUN CLOCK AT 100 KHZ ONLY, TIME
                                        MESSAGE ONCE EVERY HOUR

                                SR1=0   DEFAULT, RUN CLOCK AT ALL 3 FREQ.
                                        (60, 10 KHZ, 100 KHZ), NO TIME MESSAGES

        NOTE:   NO TIME MESSAGES CAN BE PRINTED IF THE CLOCK MODULE IS INACTIVE.
                FOR EXAMPLE, IF THE MODULE IS WAITING TO BE RELOCATED OR
                HAS BEEN DESELECTED, NO TIME MESSAGES WILL BE PRINTED DURING
                THIS PERIOD.

9.      NON-STANDARD PRINTOUTS:
        -----------------------

          ALL PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED
        IN THE DEC/X11 DOCUMENT.
```

```
              ;KW11P DEC/X11 EXERCISER MODULE

    000000'               IOMOD    <KWBJ >,172540,104,6,,,50.,12
    000000'               MODULE   140000,KWBJ ,172540,104,6,,,50.,12
                          .TITLE   KWBJ DEC/X11 SYSTEM EXERCISER MODULE
              ;           DDXCOM   VERSION  6        23-MAY-78
                                                     .LIST    BIN
                                            ;*********************************************************************
    000000'                                 BEGIN:
    000000' 053513   045102     040         MODNAM:  .ASCII   /KWBJ / ;MODULE NAME.
    000005'    000                          XFLAG:   .BYTE    OPEN                ;USED TO KEEP TRACK OF WBUFF USAGE
    000006' 172540                          ADDR:    172540+0                    ;1ST DEVICE ADDR.
    000010' 000104                          VECTOR:  104+0                       ;1ST DEVICE VECTOR.
    000012'    300                          BR1:     .BYTE    PRTY6+0            ;1ST BR LEVEL.
    000013'    000                          BR2:     .BYTE    PRTY+0             ;2ND BR LEVEL.
    000014' 000001                          DVID1:   +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' 000224'                         INIT:    START                       ;MODULE START ADDR.
    000032' 000224'                         SPOINT:  MODSP                       ;MODULE STACK POINTER.
    000034' 000000                          PASCNT:  0                           ;PASS COUNTER.
    000036' 000062                          ICONT:   50.                         ;# OF ITERATIONS PER PASS=50.
    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 MACRC 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 ERRCR
    000106' 000000                          ASB:     OPEN                        ;EXPECTED DATA.
    000110' 000000                          AWAS:    OPEN                        ;ACTUAL DATA.
    000112' 001142'                         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' 000012                          IDNUM:   12                          ;MODULE IDENTIFICATION NUMBER=12
            000040                                   .REPT    SPSIZ              ;MODULE STACK STARTS HERE.
                                                     .NLIST
                                                     .WORD    0
                                                     .LIST
                                                     .ENDR
    000224'                                 MODSP:
                                            ;*********************************************************************
```

```

                                  ;THIS MODULE EXERCISES THE KW11P PROGRAMMABLE LINE CLOCK
                                  ;AT 100KHZ,10KHZ, AND LINE FREQUENCY RATES. IT DOES
                                  ;NOT CHECK THE EXTERNAL RATE FEATURE.


                                  ;INITIALIZATION ROUTINE

                                           .GLOBL  KW11P,HMS,CKHUNG,CLOCK,PCLEAR

000224' 012767   000062  177604   START:   MOV     #50.,ICONT       ;MUST DO IN CASE OF RESTART
000232' 005767   177576                    TST     PASCNT           ;PWR FAIL RESTART?
000236' 001014                             BNE     2$               ;BR PAST CLEARS IF YES
000240' 004767   001052                    JSR     PC,PCLEAR        ;CLEAR PASS TIME TABLE
                                                                    ;FOR DATA PASSING TO MONITOR
000244' 132767   000002  000000G           BITB    #BIT1,CLOCK      ; IS KW11-P THE SYSTEM CLOCK ?
000252' 001404                             BEQ     1$               ; NO, CONTINUE
000254' 122767   000060  177522            CMPB    #'0,MODNAM+4     ; YES, IS THIS MODULE THE SYSTEM CLOCK ?
000262' 001402                             BEQ     2$               ; YES, CONTINUE
000264' 005067   001062           1$:      CLR     GOGO             ; NO, MAKE SURE CLOCK IS OFF DURING EOP

000270' 012767   000001  001062   2$:      MOV     #1,RUNING        ;SHOW WE ARE NOT BETWEEN PASSES
000276' 016701   177514                    MOV     SR1,R1           ;WHAT IS MSG INTERVAL?
000302' 006201                             ASR     R1               ;GET RID OF 50/60 HZ BIT
000304' 012767   000454  001044            MOV     #300.,INTER      ;IS IT 5 MINUTES?
000312' 022701   000001                    CMP     #1,R1
000316' 001411                             BEQ     3$               ;BR IF YES
000320' 062767   001130  001030            ADD     #600.,INTER      ;NO, SO ADD 10 MORE MINUTES
000326' 022701   000002                    CMP     #2,R1            ;IS IT 15 MINUTES?
000332' 001403                             BEQ     3$               ;BR IF YES
000334' 066767   005214  001014            ADD     2700.,INTER      ;NO, MUST BE 60 MINUTES
000342' 016767   001010  001004   3$:      MOV     INTER,MSGTIM     ;SO ADD 45 MORE MINUTES
000350' 005067   001006                    CLR     MTIME            ;MTIME IS COUNT FOR WHEN TO TYPE MSG
000354' 016701   177426                    MOV     ADDR,R1          ;GET CLOCK ADDRESS
000360' 005721                             TST     (R1)+            ;MOV IT TO THE COUNTER REGISTER
000362' 010167   001012                    MOV     R1,COUNTR        ;SAVE THIS ADDRESS
000366' 016767   177414  001000            MOV     ADDR,CLKCSR      ;SAVE CLOCK CSR ADDRESS
000374' 016700   177410                    MOV     VECTOR,R0        ;GET VECTOR LOCATION
000400' 012720   000522'                   MOV     #KLKINT,(R0)+    ;PUT INT. ROUTINE'S ADDR THERE
000404' 116720   177402                    MOVB    BR1,(R0)+        ;PUT BR LEVEL THERE TOO
000410' 016701   177402                    MOV     SR1,R1           ; COPY SR1 INTO REG. 1
000414' 042701   000007                    BIC     #7,R1            ; GET RID OF BITS 0-2
000420' 012767   000062  000744            MOV     #50.,BASE        ;ASSUME ITS A 50HZ MACHINE
000426' 032767   000001  177362            BIT     #1,SR1           ;IS IT REALLY?
000434' 001006                             BNE     4$               ;BR AROUND IF YES
000436' 062767   000012  000726            ADD     #10.,BASE        ;OTHERWISE ADD 10 TO = 60HZ
000444' 062767   000012  177364            ADD     #10.,ICONT       ;FOR 60 CPS, ICONT=60
000452' 016767   000714  000704   4$:      MOV     BASE,TIMCTR      ;NEED THIS COUNT FOR MSG COUNTER TOO
000460' 016767   177352  177432            MOV     ICONT,INTR       ;INTR ALWAYS=ICONT
000466' 012767   000001  000706            MOV     #1,COUNT         ; LOAD COUNT TO 1
000474' 012767   000115  000702            MOV     #115,COMAND      ; LOAD COMMAND FOR LINE FREQ.

000502' 016777   000674  000670   CLKGO:   MOV     COUNT,@COUNTR    ;SET UP CLOCK COUNT REGISTER
000510' 016777   000670  000656            MOV     COMAND,@CLKCSR   ;AND START IT OFF
000516' 104400   000000'                   EXIT$,BEGIN                      ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```

```

                                  ;INTERRUPT SERVICE ROUTINE

000522' 005367   000636           KLKINT:  DEC     TIMCTR           ; HAS 1 SECOND ELAPSED ?
000526' 001113                             BNE     9$               ;BR IF NOT
000530' 005077   000640                    CLR     @CLKCSR          ; DISABLE FURTHER INTERRUPTS
000534' 062767   000001  000712            ADD     #1,TIME          ;COUNT A SECOND FOR TOTAL TIME
000542' 005567   000710                    ADC     TIMEXT           ; PUT ANY OVERFLOW TIME HERE
000546' 005267   000610                    INC     MTIME            ;COUNT A SECOND FOR MSG TIME
000552' 005767   000602                    TST     RUNING           ;ARE WE BETWEEN PASSES?
000556' 001021                             BNE     1$               ;NO CONTINUE
000560' 016767   000606  000576            MOV     BASE,TIMCTR      ; YES, RESET THE INTERRUPT COUNTER
000566' 016767   000662  000572            MOV     TIME,TMPTIM      ;GET THE PRESENT TIME
000574' 166767   000570  000564            SUB     HLDTIM,TMPTIM    ;GET ELAPSED TIME SINCE EOP
000602' 022767   001604  000556            CMP     #900.,TMPTIM     ;HAVE 15 MINUTES PASSED?
000610' 101062                             BHI     9$               ;NO, CONTINUE
000612' 062767   001604  000550            ADD     #900.,HLDTIM     ;YES, CHECK AGAIN IN 15 MINUTES
000620' 000471                             BR      11$              ;GO CHECK FOR ANY "HUNG" MODULES
000622'                           1$:
                                           ;-------------------------------------------------------------------------
000622' 000004   000000' 000630'           PIRQ$,BEGIN,2$           ; QUEUE UP TO CONTINUE AT 2$ AND RTI
                                           ;-------------------------------------------------------------------------
000630' 032701   000030           2$:      BIT     #30,R1           ; LOCK AT ANY PARTICULAR FREQ. ?
000634' 001407                             BEQ     3$               ; NO, CONTINUE
000636' 022701   000030                    CMP     #30,R1           ; LOCK AT 100 KHZ ?
000642' 001411                             BEQ     4$               ; GO DO IT
000644' 032701   000020                    BIT     #20,R1           ; LOCK AT 10 KHZ ?
000650' 001015                             BNE     5$               ; YES, GO DO IT
000652' 000423                             BR      6$               ; LOCK AT LINE FREQ., DO IT

000654' 022767   000246  000520   3$:      CMP     #166.,COUNT      ; FIND OUT WHAT PRESENT COUNT IS AT
000662' 101010                             BHI     5$               ; COUNT NOW = 1
000664' 103416                             BLO     6$               ; COUNT NOW = 1666.
000666' 012767   003202  000506   4$:      MOV     #1666.,COUNT     ; COUNT WAS 166., NOW IS 1666.
000674' 012767   000101  000502            MOV     #101,COMAND      ; SET FREQ. TO 100 KHZ
000702' 000421                             BR      7$               ; CONTINUE

000704' 012767   000246  000470   5$:      MOV     #166.,COUNT      ; COUNT IS NOW 166.
000712' 012767   000103  000464            MOV     #103,COMAND      ; SET FREQ. TO 10 KHZ
000720' 000412                             BR      7$               ; CONTINUE
000722' 012767   000001  000452   6$:      MOV     #1,COUNT         ; COUNT IS NOW 1
000730' 012767   000115  000446            MOV     #115,COMAND      ; SET FREQ. TO LINE FREQ.
000736' 016767   000430  000420            MOV     BASE,TIMCTR      ; SETUP # OF INTERUPTS FOR 50/60 HZ
000744' 000403                             BR      8$               ; CONTINUE

000746' 012767   000074  000410   7$:      MOV     #60.,TIMCTR      ; 60. INTERUPTS = 1 SECOND
000754' 000434                    8$:      BR      13$              ; GO START CLOCK

000756' 032777   000010  000410   9$:      BIT     #BIT3,@CLKCSR    ; IN THE REPEAT INTERRUPT MODE ?
000764' 001006                             BNE     10$              ; YES, GET OUT
000766' 016777   000410  000404            MOV     COUNT,@COUNTR    ;IF NOT, LOAD THE COUNTER
000774' 016777   000404  000372            MOV     COMAND,@CLKCSR   ; TURN ON THE CLOCK
001002' 000002                    10$:     RTI                      ; RETURN
```

```

001004'                            11$:
                                           ;-----------------------------------------------------------------
001004' 000004  000000' 001012'            PIRQ$,BEGIN,12$             ; QUEUE UP TO CONTINUE AT 12$ AND RTI
                                           ;-----------------------------------------------------------------
001012' 012777  000001  000360     12$:    MOV     #1,@COUNTR          ;SET COUNT
001020' 016777  000326  000346             MOV     GOGO,@CLKCSR        ;TURN THE CLOCK ON IF SYSTEM CLOCK
001026' 010546                             MOV     R5,-(SP)            ;SAVE R5
001030' 012705  002110'                    MOV     #R5STK,R5           ;SET UP R5 STACK
001034' 004767  000000G                    JSR     PC,CKHUNG           ; BACK TO MONITOR. CHECK FOR HUNG MODULES
001040' 012605                             MOV     (SP)+,R5            ;RESTORE R5
001042' 104400  000000'                    EXIT$,BEGIN                          ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

001046' 016700  000320             13$:    MOV     BASE,R0             ;SAVE THE BASE TIME
001052' 005300                             DEC     R0                  ;SETUP R0
001054' 026700  176760                     CMP     ICOUNT,R0           ;IS IT TIME FOR END PASS
001060' 001024                             BNE     14$                 ;NO, BRANCH
001062' 005067  000272                     CLR     RUNING              ;QUIT COUNTING PASS SECONDS
001066' 016767  000362  000274             MOV     TIME,HLDTIM         ;SAVE THE TIME OF THE EOP
001074' 012767  000001  000300             MOV     #1,COUNT            ; SET THE COUNT
001102' 016767  000264  000254             MOV     BASE,TIMCTR         ; LOAD NUMBER OF INTERRUPTS COUNTER
001110' 012767  000115  000266             MOV     #115,COMAND         ; LOAD THE COMMAND
001116' 016777  000260  000254             MOV     COUNT,@COUNTR       ;SET COUNT
001124' 016777  000222  000242             MOV     GOGO,@CLKCSR        ;KEEP CLOCK GOING IF SYSTEM CLOCK

001132'                            14$:
001132' 104413  000000'                    ENDIT$,BEGIN                ;SIGNAL END OF ITERATION.
                                                                       ;MONITOR SHALL TEST END OF PASS
001136' 000167  177340                     JMP     CLKGO               ;IF NOT, KEEP ON CLOCKIN'

001142' 005767  176666             RESTRT: TST     PASCNT              ; THIS IS FOR CSS BUS SWITCHES
001146' 001002                             BNE     REST                ; CONTINUE
001150' 000167  177050                     JMP     START               ; BEGIN AT START
001154' 026767  000202  000174     REST:   CMP     MTIME,INTER         ;IS IT TIME FOR A MSG?
001162' 103450                             BLO     1$                  ;NO, SO BRANCH
001164' 005067  000172                     CLR     MTIME               ;YES, RESET MSG TIME COUNT
001170' 026727  176622  000002             CMP     SR1,#2              ;IS ANY MSG WANTED?
001176' 002442                             BLT     1$                  ;BR IF NOT

                                   ;+
                                   ; SET UP R5 STACK AND CONVERT TIME AND TIMEXT TO HOURS, MINUTES AND SECONDS.
                                   ;                 CALLING SEQUENCE: CALL HMS IN <TIME,TIMEXT> OUT <R0>
                                   ;-

001200' 010046                             MOV     R0,-(SP)            ;SAVE R0
001202' 010546                             MOV     R5,-(SP)            ;SAVE R5
001204' 012705  002110'                    MOV     #R5STK,R5           ;SET UP R5 STACK
001210' 162705  000002                     SUB     #1*#2,R5            ;SAVE SPACE ON STACK FOR OUTPUT
                                                                       ;ARGUMENT FOR HMS CALL
001214' 010546                             MOV     R5,-(SP)            ;SAVE R5 STACK POINTER ON R6 STACK
001216' 016745  000234                     MOV     TIMEXT,-(R5)        ;PLACE EXTENDED ELAPSED TIME
                                                                       ;ARGUMENT ON R5 STACK
001222' 016745  000226                     MOV     TIME,-(R5)          ;PLACE ELAPSED TIME ARGUMENT
                                                                       ;ON R5 STACK
001226' 004767  000000G                    JSR     PC,HMS              ;CALL HOURS, MINUTES, SECOND CONVERSION
001232' 012605                             MOV     (SP)+,R5            ;RESTORE R5 STACK POINTER
```

```
001234' 012500                             MOV     (R5)+,R0            ;GET OUTPUT ARGUMENT OF HMS
                                                                       ;ROUTINE INTO R0
001236' 112067  000166                     MOVB    (R0)+,RUNT1+1       ;LOAD
001242' 112067  000163                     MOVB    (R0)+,RUNT1+2       ;CONVERTED
001246' 112067  000160                     MOVB    (R0)+,RUNT1+3       ;TIME
001252' 112067  000155                     MOVB    (R0)+,RUNT2         ;INTO
001256' 112067  000152                     MOVB    (R0)+,RUNT2+1       ;MESSAGE
001262' 112067  000150                     MOVB    (R0)+,RUNT3
001266' 112067  000145                     MOVB    (R0)+,RUNT3+1
001272' 012605                             MOV     (SP)+,R5            ;RESTORE R5
001274' 012600                             MOV     (SP)+,R0            ;RESTORE R0
001276' 104403  000000' 001346'            MSGN$,BEGIN,EXPRI           ;ASCII MESSAGE CALL WITH COMMON HEADER
001304' 012767  000001  000046     1$:     MOV     #1,RUNING           ; SHOW WE ARE NOT BETWEEN PASSES
001312' 000167  177164                     JMP     CLKGO               ; GO RESTART THE CLOCK

001316' 012700  001460'            PCLEAR: MOV     #MODTIM,R0          ;GET START OF TABLE
001322' 012701  002040'                    MOV     #MODEND,R1          ;GET END
001326' 005067  000122                     CLR     TIME                ; ZERO TOTAL RUNTIME
001332' 005067  000120                     CLR     TIMEXT              ; ZERO EXTENDED TIME BITS
001336' 005020                     1$:     CLR     (R0)+               ;CLEAR ENTRY
001340' 020001                             CMP     R0,R1               ;DONE?
001342' 001375                             BNE     1$                  ;BR IF NOT
001344' 000207                             RTS     PC                  ;EXIT
```

```
416
417
418
419  001346' 001406'                       EXPRI:  RUNTI
420  001350' 177777                                177777
421
422  001352' 000115                        GOGO:   115                   ; IF THIS IS THE SYSTEM CLOCK, ELSE = 0
423  001354' 000000                        MSGTIM: OPEN                  ;TELLS WHEN TO TYPE A MSG
424  001356' 000000                        INTER:  OPEN                  ;HOW OFTEN BETWEEN MSG.S
425  001360' 000000                        RUNING: OPEN                  ;0 IF BETWEEN PASSES
426  001362' 000000                        MTIME:  OPEN                  ;COUNTS SECONDS BETWEEN MSG.S
427  001364' 000000                        TIMCTR: OPEN                  ;COUNTS # OF INTERRUPTS UP TO 1 SECOND
428  001366' 000000                        TMPTIM: OPEN                  ;TEMP LOCATION HOLDS TIME
429  001370' 000000                        HLDTIM: OPEN                  ;TEMP LOCATION HOLDS TIME SINCE LAST EOP
430  001372' 000000                        BASE:   OPEN                  ; HOLDS LINE FREQ.,  50./60.
431  001374' 000000                        CLKCSR: OPEN                  ; HOLDS ADDRESS OF CLOCK CSR
432  001376' 000000                        CLKCTR: OPEN                  ; HOLDS ADDRESS OF CLOCK COUNT SET
433  001400' 000000                        COUNTR: OPEN                  ; HOLDS ADDRESS OF CLOCK COUNTER
434  001402' 000001                        COUNT:  1                     ; NUMBER OF CLOCK TICKS BEFORE INTERRUPT
435  001404' 000115                        COMAND: 115                   ; COMMAND TO THE CLOCK
436
437
438  001406' 020045  052522  020116  RUNTI:  .ASCII '% RUN TIME IS     '
439  001414' 044524  042515  044440
440  001422' 020123  020040     040
441  001427'    040  020040     072  RUNT1:  .ASCII  '   :'      ; HOURS
442  001433'    040  035040          RUNT2:  .ASCII  '  :'       ; MINUTES
443  001436' 020040  024040  035110  RUNT3:  .ASCIZ  '   (H:M:S)%'  ; SECONDS
444  001444' 035115  024523  000045
445                                  ;
446                                  ;       DO NOT MODIFY ANY LOCATION BELOW THIS POINT.
447                                  ;       THE MONITOR MUST KNOW WHERE EACH WORD IS.
448                                  ;
449                                  .EVEN
450  001452' 000000'                 KW11P:  BEGIN                 ; HOLDS STARTING ADDRESS OF MODULE
451  001454' 000000                  TIME:   OPEN                  ;ELAPSED TIME IN SECONDS
452  001456' 000000                  TIMEXT: OPEN                  ; HOLDS EXTENDED BITS FOR THE TIME WORD
453  001460' 000170                  MODTIM: .BLKW   120.    ; TABLE FOR MONITOR, HOLDS TIME FOR EACH MODULE
454  002040'                         MODEND:
455
456                                  ;+
457                                  ; SET ASIDE R5 STACK STORAGE
458                                  ;-
459
460  002110' 000000                  R5STK:  .WORD   0             ;R5 STACK
461
462          000001                          .END
```

```
ACSR      000102R      211#
ADDR      000006R      177#    262     265
ADDR22=   001000       229#
ASB       000106R      215#
ASTAT     000104R      213#
AWAS      000110R      216#
BASE      001372R      271*    274*    276     295     326     350     357     430#
BEGIN     000000R      174#    283     304     340     348     363     404     450"
BIT0   =  000001       229#
BIT1   =  000002       229#    244
BIT10  =  002000       229#
BIT11  =  004000       229#
BIT12  =  010000       229#
BIT13  =  020000       229#
BIT14  =  040000       229#
BIT15  =  100000       229#
BIT2   =  000004       229#
BIT3   =  000010       229#    332
BIT4   =  000020       229#
BIT5   =  000040       229#
BIT6   =  000100       229#
BIT7   =  000200       229#
BIT8   =  000400       229#
BIT9   =  001000       229#
BREAK$=   104407       229#
BR1       000012R      179#    268
BR2       000013R      180#
BTOD$  =  104421       229#
CDATA$=   104412       229#
CKHUNG=   ****** G     237#    346
CLKCSR    001374R      265*    282*    289*    332     335*    343*    360*    431#
CLKCTR    001376R      432#
CLKGO     000502R      281#    365     406
CLOCK  =  ****** G     237#    244
COMAND    001404R      279*    282     318*    322*    325*    335     358*    435#
CONFIG    000056R      199#
COUNT     001402R      278*    281     314     317*    321*    324*    334     356*    359     434#
COUNTR    001400R      264*    281*    334*    342*    359*    433#
CSRA      000100R      209#
DATCK$=   104411       229#
DATER$=   104404       229#
DVID1     000014R      181#
ENDIT$=   104413       229#    363
END$   =  104410       229#
ERRTYP    000106R      214#
EXIT$  =  104400       229#    283     348
EXPRI     001346R      404     419#
GETPA$=   104415       229#
GOGO      001352R      248*    343     360     422#
GWBUF$=   104414       229#
HLDTIM    001370R      297     300*    355*    429#
HMS    =  ****** G     237#    391
HRDCNT    000044R      194#
HRDER$=   104405       229#
HRDPAS    000050R      196#
ICONT     000036R      191#    239*    275*    277
```

```
ICOUNT  000040R                 192#    352
IDNUM   000122R                 221#
INIT    000030R                 188#
INTER   001356R                 253*    256*    259*    260     370     424#
INTR    000120R                 220#    277*
KLKINT  000522R                 267     287#
KW11P   001452RG                237#    450#
MAP22$= 104416                  229#
MODEND  002040R                 409     454#
MODNAM  000000R                 175#    246
MODSP   000224R                 189     227#
MODTIM  001460R                 408     453#
MSGN$ = 104403                  229#    404
MSGS$ = 104402                  229#
MSGTIM  001354R                 260*    423#
MSG$  = 104401                  229#
MTIME   001362R                 261*    292*    370     372*    426#
NULL  = 000000                  229#
OPEN  = 000000                  176     182     183     184     185     202     203     204     205     206     207     208     209
                                211     213     215     216     218     219     220     229#    423     424     425     426     427
                                428     429     430     431     432     433     451     452

OTOA$ = 104420                  229#
PASCNT  000034R                 190#    240     367
PCLEAR  001316RG                237#    242     408#
PIRQ$ = 000004                  229#    304     340
POPSP = 005726                  229#
POPSP2= 022626                  229#
PRTY  = 000000                  180     229#
PRTY0 = 000000                  229#
PRTY1 = 000040                  229#
PRTY2 = 000100                  229#
PRTY3 = 000140                  229#
PRTY4 = 000200                  229#
PRTY5 = 000240                  229#
PRTY6 = 000300                  179     229#
PRTY7 = 000340                  229#
PS    = 177776                  229#
PSW   = 177776                  229#
PUSH  = 005746                  229#
PUSH2 = 024646                  229#
RAND$ = 104417                  229#
RANNUM  000054R                 198#
REST    001154R                 368     370#
RESTRT  001142R                 217     367#
RES1    000056R                 200#
RES2    000060R                 201#
RSTRT   000112R                 217#
RUNING  001360R                 250*    293     354*    405*    425#
RUNTI   001406R                 419     438#
RUNT1   001427R                 395*    396*    397*    441#
RUNT2   001433R                 398*    399*    442#
RUNT3   001436R                 400*    401*    443#
R5STK   002110R                 345     383     460#
SBADR   000102R                 210#
SOFCNT  000042R                 193#
SOFER$= 104406                  229#
```

```
SOFPAS  000046R                 195#
SPOINT  000032R                 189#
SPSIZ = 000040                    1#    222
SR1     000016R                 182#    251     269     272     373
SR2     000020R                 183#
SR3     000022R                 184#
SR4     000024R                 185#
START   000224R                 188     239#    369
STAT    000026R                 187#
SVR0    000062R                 202#
SVR1    000064R                 203#
SVR2    000066R                 204#
SVR3    000070R                 205#
SVR4    000072R                 206#
SVR5    000074R                 207#
SVR6    000076R                 208#
SYSCNT  000052R                 197#
TIMCTR  001364R                 276*    287*    295*    326*    329*    357*    427#
TIME    001454R                 290*    296     355     389     410*    451#
TIMEXT  001456R                 291*    387     411*    452#
TMPTIM  001366R                 296*    297*    298     428#
TRPDFD= 000022                  229#
VECTOR  000010R                 178#    266
WASADR  000104R                 212#
WDFR    000116R                 219#
WDTO    000114R                 218#
XFLAG   000005R                 176#
.     = 002112R                 453#

. ABS.  000000      000
        002112      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XKWBJO,XKWBJO/SOL/CRF:SYM=DDXCOM,XKWBJO
 RUN-TIME: 1 1 .2 SECONDS
 RUN-TIME RATIO: 28/3=8.3
 CORE USED:  7K  (13 PAGES)
```