

-REM -

IDENTIFICATION

PRODUCT CODE: AC-E691J-MC
PRODUCT NAME: CXKWBJO 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 CONSIDERED 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 GATE FEATURE. WHENEVER LINE FREQUENCY IS SELECTED, IT IS RUN IN THE REPEAT INTERRUPT MODE. OTHER FREQUENCIES ARE ALWAYS RUN IN THE 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
5	50	15
6	60	60
7	50	60
10	50/60	
20	10,000	
30	100,000	

; ALL TIMES ARE APPROXIMATE

RUN CLOCK AT LINE FREQ. ONLY
(MUST ALSO SET BIT0=1 FOR 50 HZ)
RUN CLOCK AT 10 KHZ ONLY
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.

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

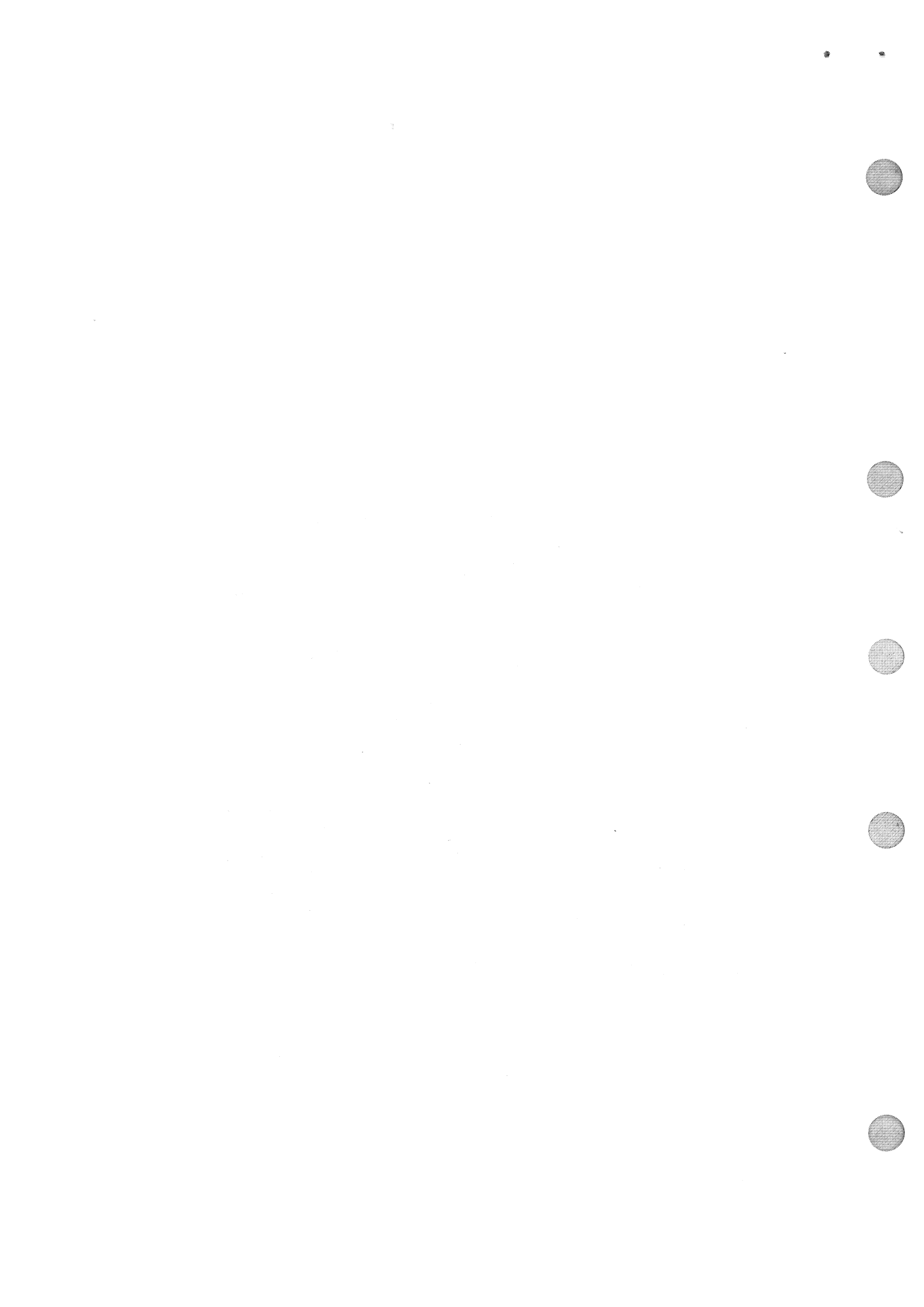
SR1=0 DEFAULT, RUN CLOCK AT ALL 3 FREQ.
(60, 10 KHZ, 100 KHZ), NO TIME MESSAGES

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.



JKW11P 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 LIST BIN
*****
000000* BEGIN:
000000* 053513 045102 040 MODNAM: ASCII /KWBJ /MODULE NAME
000000* 000 XFLAG: BYTE OPEN ;USED TO KEEP TRACK OF WBUF USAGE
000000* 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* 800 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 INT: START ;MODULE START ADDR.
000032* 000224 SPOINT: MODDSP ;MODULE STACK POINTER.
000034* 000000 PASCNT: 0 ;PASS COUNTER.
000036* 000062 ICOUNT: 50. ;# OF ITERATIONS PER PASS=50.
000040* 000000 SOFCNT: 0 ;LDC TO COUNT REPEATONS
000042* 000000 HRDCNT: 0 ;LDC TO SAVE TOTAL SOFT ERRORS
000044* 000000 SOFPAS: 0 ;LDC TO SAVE TOTAL HARD ERRORS
000046* 000000 HRDPAS: 0 ;LDC TO SAVE SOFT ERRORS PER PASS
000050* 000000 SVSCNT: 0 ;LDC TO SAVE HARD ERRORS PER PASS
000052* 000000 RANNUM: 0 ;# OF SYS ERRORS ACCUMULATED
000054* 000000 CONFIG: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056* 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000060* 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000062* 000000 SVR0: OPEN ;LDC TO SAVE R0.
000064* 000000 SVR1: OPEN ;LDC TO SAVE R1.
000066* 000000 SVR2: OPEN ;LDC TO SAVE R2.
000070* 000000 SVR3: OPEN ;LDC TO SAVE R3.
000072* 000000 SVR4: OPEN ;LDC TO SAVE R4.
000074* 000000 SVR5: OPEN ;LDC TO SAVE R5.
000076* 000000 SVR6: OPEN ;LDC TO SAVE R6.
000100* 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
000102* SBADR: ;ADDR OF GOOD DATA, OR
000104* 000000 ACSR: OPEN ;CONTENTS OF CSR.
000106* 000000 MBSADR: ;ADDR OF BAD DATA, OR
000108* 000000 ASADR: OPEN ;STATUS REG CONTENTS.
000110* 000000 FRRTYP: ;TYPE OF ERROR
000112* 000000 ASB: OPEN ;EXPECTED DATA.
000114* 000000 AMAS: OPEN ;ACTUAL DATA.
000116* 001142 RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
000118* 000000 WDO: OPEN ;WORDS TO MEMORY PER ITERATION
000120* 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
000122* 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
```

```
000122* 000012 IDNUM: 12 ;MODULE IDENTIFICATION NUMBFF=12
000040* 000040 .REPT SPSIZ ;MODULE STACK STARTS HERE.
; .LIST 0
; .LIST
; .ENDR
000224* MODDSP:
*****
```

```

229 ;THIS MODULE EXERCISES THE KW11P PROGRAMMABLE LINE CLOCK
230 ;AT 100KHZ,10KHZ AND LINE FREQUENCY RATES. IT DOES
231 ;NOT CHECK THE EXTERNAL RATE FEATURE.
232
233 ;INITIALIZATION ROUTINE
234
235 .GLOBL KW11P,HMS,CKHUNG,CLOCK,PCLEAR
236
237
238
239
240 000224* 012767 000062 177604 START: MOV #50,ICONT ;MUST DO IN CASE OF RESTART
241 000232* 005767 177576 TST PASCNT ;PWR FAIL RESTART?
242 000236* 001014 BNE Z8 ;BR PAST CLEARS IF YES
243 000240* 004767 001052 JSR PC,PCLEAR ;CLR PASS TIME TABLE
244 ;FOR DATA PASSING TO MONITOR
245 000244* 132767 000002 000000G BITB #BIT1,CLOCK ;IS KW11-P THE SYSTEM CLOCK ?
246 000252* 001404 BEQ 4$ ;NO, CONTINUE
247 000254* 127167 000060 177522 CMPB #0,MODNAM+4 ;IS THIS MODULE THE SYSTEM CLOCK ?
248 000264* 005067 001062 1$: CLR ;YES, CONTINUE
249 ; NO, MAKE SURE CLOCK IS OFF DURING EOP
250
251 000270* 012767 000001 001062 2$: MOV #1,RUNING ;SHOW WE ARE NOT BETWEEN PASSES
252 000276* 012701 177514 MOV SRI,R1 ;WHAT IS MSG INTERVAL?
253 000300* 006201 ASR R1 ;GET RID OF 50/60 HZ BIT
254 000304* 012767 000454 001044 MOV #300,INTER ;IS IT 5 MINUTES?
255 000316* 001411 CMP #1,R1 ;BR IF YES
256 000320* 062767 001130 001030 ADD #600,INTER ;NO, SO ADD 10 MORE MINUTES
257 000326* 022701 000002 2$: CMP #2,R1 ;IS IT 15 MINUTES?
258 000334* 062767 005214 001014 ADD #700,INTER ;NO, MUST BE 60 MINUTES
259 000344* 016767 001010 001004 3$: MOV INTER,MSGTIM ;SO ADD 45 MORE MINUTES
260 000354* 005067 001006 CLR ;MTIME IS COUNT FOR WHEN TO TYPE MSG
261 000366* 005767 177426 MOV ADDR,R1 ;GET CLOCK ADDRESS
262 000374* 016767 177410 001000 MOV R1,COUNTN ;MOV IT TO THE COUNTER REGISTER
263 000386* 005767 177426 TST ;SAVE THIS ADDRESS
264 000396* 016767 177414 MOV ADDR,CLKCSR ;SAVE CLOCK CSR ADDRESS
265 000404* 012720 000522 MOV VECTOR,R0 ;GET VECTOR LOCATION
266 000410* 016701 177402 MOVB BRI,(R0)+ ;PUT BR LEVEL THERE TOO
267 000420* 012701 000007 MOV SRI,R1 ;COPY SRI INTO REG-1
268 000426* 032767 000001 000744 BIC #0,BASE ;GET RID OF BITS 0-2
269 000434* 001006 MOV #1,SRI ;ASSUME IT'S A 50HZ MACHINE
270 000444* 062767 000012 000726 BIT ;IS IT REALLY?
271 000452* 016767 000714 000704 4$: RNE ;BR AROUND IF YES
272 000460* 016767 177352 177432 ADD #10,BASE ;OTHERWISE ADD 10 TO = 60HZ
273 000466* 012767 000012 000706 MOV #1,ICONT ;OTHERWISE COUNT=60
274 000474* 012767 000015 000702 MOV #1,COUNT ;NEED THIS COUNT FOR MSG COUNTER TOO
275 000480* 016767 177352 177432 MOV #1,INTR ;INTR ALWAYS=ICONT
276 000486* 012767 000015 000702 MOV #1,COUNT ;LOAD COUNT TO 1
277 000494* 012767 000015 000702 MOV #15,COMAND ;LOAD COMMAND FOR LINE FREQ.
278
279 000502* 016777 000674 000670 CLKG0: MOV COUNT,@COUNTN ;SET UP CLOCK COUNT REGISTER
280 000510* 016777 000670 000656 MOV COMAND,@CLKCSR ;AND START IT OFF
281 000516* 104400 000000 EXIT$,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

```

```

284 ;INTERRUPT SERVICE ROUTINE
285
286
287 000522* 005367 000636 KLKINT: DEC TIMCTR ; HAS 1 SECOND ELAPSED ?
288 000526* 001113 BNE 9$ ;BR IF NOT
289 000530* 005977 000640 CLR @CLKCSR ;DISABLE FURTHER INTERRUPTS
290 000534* 016767 000001 000712 ADD #1,TIMCTR ;COUNT A SECOND FOR TOTAL TIME
291 000542* 005567 000710 ADC TIMEXT ;PUT ANY OVERFLOW TIME HERE
292 000546* 005267 000610 INC MTIME ;COUNT A SECOND FOR MSG TIME
293 000552* 005767 000602 TST RUNING ;ARE WE BETWEEN PASSES?
294 000556* 016767 000606 BNE 1$ ;NO, CONTINUE
295 000560* 016767 000606 000576 MOV BASE,TIMCTR ;YES, RESET THE INTERRUPT COUNTER
296 000566* 016767 000662 000572 MOV TIME,TMPTIM ;GET THE PRESENT TIME
297 000574* 166767 000570 SUB HLDTIM,TMPTIM ;GET ELAPSED TIME SINCE EOP
298 000602* 022767 001604 000556 CMP #900,TMPTIM ;HAVE 15 MINUTES PASSED?
299 000610* 101062 BHI 8$ ;NO, CONTINUE
300 000612* 062767 001604 000550 ADD #900,HLDTIM ;YES, CHECK AGAIN IN 15 MINUTES
301 000620* 000471 BR 11$ ;GO CHECK FOR ANY "HUNG" MODULES
302
303
304 000622* 000004 000000* 000630* 1$: FIRQ$,BEGIN,2$ ;QUEUE UP TO CONTINUE AT 2$ AND RTI
305
306 000630* 032701 000030 2$: BIT #30,R1 ;LOCK AT ANY PARTICULAR FREQ. ?
307 000634* 001407 BNE 4$ ;NO, CONTINUE
308 000636* 022701 000030 3$: CMP #30,R1 ;LOCK AT 100 KHZ ?
309 000642* 001411 BNE 4$ ;GO DO IT
310 000644* 032701 000020 4$: BIT #20,R1 ;LOCK AT 10 KHZ ?
311 000650* 001015 BNE 5$ ;YES, GO DO IT
312 000652* 000423 BR 6$ ;LOCK AT LINE FREQ., DO IT
313
314 000654* 022767 000246 000520 3$: CMP #166,COUNT ;FIND OUT WHAT PRESENT COUNT IS AT
315 000662* 101010 BHI 6$ ;COUNT NOW = 1
316 000664* 103416 BLO 6$ ;COUNT NOW = 1666.
317 000666* 012767 003202 000506 4$: MOV #1666,COUNT ;COUNT WAS 166, NOW IS 1666.
318 000674* 012767 000101 000502 MOV #101,COMAND ;SET FREQ. TO 100 KHZ
319 000702* 000421 BR ;CONTINUE
320
321 000704* 012767 000246 000470 5$: MOV #166,COUNT ;COUNT IS NOW 166.
322 000712* 012767 000103 000464 MOV #103,COMAND ;SET FREQ. TO 10 KHZ
323 000720* 000412 BR ;CONTINUE
324 000722* 012767 000001 000452 6$: MOV #1,COUNT ;COUNT IS NOW 1
325 000730* 012767 000115 000446 MOV #15,COMAND ;SET FREQ. TO LINE FREQ.
326 000736* 016767 000430 000420 MOV BASE,TIMCTR ;SETUP # OF INTERRUPTS FOR 50/60 HZ
327 000744* 000403 BR ;CONTINUE
328
329 000746* 012767 000074 000410 7$: MOV #60,TIMCTR ;60. INTERRUPTS = 1 SECOND
330 000754* 000434 BR 8$ ;GO START CLOCK
331
332 000756* 032777 000010 000410 9$: BIT #BIT3,@CLKCSR ;IN THE REPEAT INTERRUPT MODE ?
333 000764* 001006 BNE 10$ ;YES, GET OUT
334 000766* 016777 000410 000404 MOV COUNT,@COUNTN ;IF NOT, LOAD THE COUNTER
335 000774* 016777 000404 000372 MOV COMAND,@CLKCSR ;TURN ON THE CLOCK
336 001002* 000002 RTI ;RETURN

```

```

337 001004* 11$:
38
39
40 001004* 000004 000000* 001012* ;PIRQS,BEGIN,12$ ; QUEUE UP TO CONTINUE AT 12$ AND RTI
41
42 001012* 012777 000001 000360 12$: MOV #1,COUNTR ;SET COUNT
43 001020* 012777 000326 000346 MOV GDC,CLKCSR ;TURN THE CLOCK ON IF SYSTEM CLOCK
44 001026* 010546 MOV R5,-(SP) ;SAVE R5
45 001030* 012705 002110* MOV #R5,RS ;SET UP R5 STACK
46 001034* 004767 000000G JSR PC,CRHUNG ;BACK TO MONITOR. CHECK FOR HUNG MODULES
47 001040* 012605 MOV (SP)+,R5 ;RESTORE R5
48 001042* 104400 000000* EXIT$,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
49
50 001046* 016700 000320 13$: MOV BASE,R0 ;SAVE THE BASE TIME
51 001052* 005300 DEC R0 ;SETUP R0
52 001054* 026700 176760 CMP ICOUNT,R0 ;IS IT TIME FOR END PASS
53 001060* 001024 BNE 14$ ;NO, BRANCH
54 001062* 005067 000272 CLR RUNING ;QUIT COUNTING PASS SECONDS
55 001066* 016757 000362 MOV TIME,HLDTIM ;SAVE THE TIME OF THE EOP
56 001074* 012767 000001 000274 MOV #COUNT ;SET THE COUNT
57 001102* 016767 000264 000254 MOV BASE,TIMCTR ;LOAD NUMBER OF INTERRUPTS COUNTER
58 001110* 012767 000115 000266 MOV #15,COMAND ;LOAD THE COMMAND
59 001116* 016777 000260 000254 MOV COUNTR ;LOAD COUNT
60 001124* 016777 000222 000242 MOV GDC,CLKCSR ;KEEP CLOCK GOING IF SYSTEM CLOCK
61
62 001132* 104413 000000* 14$: ENDT$,BEGIN ;SIGNAL END OF ITERATION.
63 001132* 104413 000000* MONITOR SHALL TEST END OF PASS
64 001136* 000167 177340 JMP CLKGO ;IF NOT, KEEP ON CLOCKIN
65
66 001142* 005767 176666 RESTRT: TST PASCNT ; THIS IS FOR CSS BUS SWITCHES
67 001146* 001002 BNE REST ; CONTINUE
68 001150* 000167 177050 JMP START ; BEGIN AT START
69 001154* 026767 000202 000174 REST: CMP MTIME,INTER ; IS IT TIME FOR A MSG?
70 001162* 013450 BLD 1 ; NO, GO BRANCH
71 001164* 005067 000172 CLR MTIME ; YES, RESET MSG TIME COUNT
72 001170* 026727 176622 000002 CMP SRI,#2 ; IS ANY MSG WANTED?
73 001176* 002442 BLT 1$ ; BR IF NOT
74
75 ;+
76 ; SET UP R5 STACK AND CONVERT TIME AND TIMEXT TO HOURS, MINUTES AND SECONDS.
77 ; CALLING SEQUENCE: CALL HMS IN <TIME,TIMEXT> OUT <R0>
78 ;-
79
80
81 001200* 010046 MOV R0,-(SP) ;SAVE R0
82 001204* 019546 MOV R5,-(SP) ;SAVE R5
83 001204* 012705 MOV #R5,RS ;SET UP R5 STACK
84 001210* 162705 000002 SUB #1#,RS ;SAVE SPACE ON STACK FOR OUTPUT
85
86 001214* 010546 MOV R5,-(SP) ;SAVE R5 STACK POINTER ON R6 STACK
87 001216* 016745 000234 MOV TIMEXT,-(R5) ;PLACE EXTENDED ELAPSED TIME
88
89 001222* 016745 000226 MOV TIME,-(R5) ;ARGUMENT ON R5 STACK
90
91 001226* 004767 000000G JSR PC,HMS ;PLACE ELAPSED TIME ARGUMENT
92 001232* 012605 MOV (SP)+,R5 ;ON R5 STACK
;CALL HOURS, MINUTES, SECOND CONVERSION
;RESTORE R5 STACK POINTER

```

```

393 001234* 012500 MOV (R5)+,R0 ;GET OUTPUT ARGUMENT OF HMS
394
395 001236* 112067 000166 MOVB (R0)+,RUNT1+1 ;ROUTINE INTO R0
396 001242* 112067 000163 MOVB (R0)+,RUNT1+2 ;LOAD
397 001246* 112067 000160 MOVB (R0)+,RUNT1+3 ;CONVERTED
398 001252* 112067 000155 MOVB (R0)+,RUNT2 ;TIME
399 001256* 112067 000152 MOVB (R0)+,RUNT2+1 ;INTO
400 001262* 112067 000150 MOVB (R0)+,RUNT3 ;MESSAGE
401 001266* 112067 000145 MOVB (R0)+,RUNT3+1
402 001272* 012605 MOV (SP)+,R5 ;RESTORE R5
403 001274* 012605 MOV (SP)+,R0 ;RESTORE R0
404 001276* 104403 000000* 001346* MSGNS,BEGIN,EXPI ;ASCII MESSAGE CALL WITH COMMON HEADER
405 001304* 012767 000001 000046* 1$: MOV #1,RUNING ;SHOW WE ARE NOT BETWEEN PASSES
406 001312* 000167 177164 JMP CLKGO ;GO RESTART THE CLOCK
407
408 001316* 012700 001460* PCLEAR: MOV #MODTIM,R0 ;GET START OF TABLE
409 001322* 012701 002040* MOV #MODEND,R1 ;GET END
410 001326* 005067 000122 CLR TIME ;ZERO TOTAL RUNTIME
411 001332* 005067 000120 CLR TIMEXT ;ZERO EXTENDED TIME BITS
412 001336* 005020 1$: CLR (R0)+ ;CLEAR ENTRY
413 001340* 020001 CMP R0,R1 ;DONAR
414 001344* 013375 BNE 1$ ;BR IF NOT
415 001344* 000207 RTS PC ;EXIT

```