

.REM -

-----  
IDENTIFICATION  
-----

PRODUCT CODE: AC-F001B-MC  
PRODUCT NAME: CKKWF80 GROSS TMNG MOD  
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 OR EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1976,1978 DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT

"KWF" IS A GROSS TIMING ANALYSIS MODULE THAT USES THE KW11-L LINE CLOCK OPTION TO PERFORM AN OVERALL TIMING ANALYSIS ON THE 2040 PDP11 CONSOLE PROGRAM SYSTEM OR A DM2X SECONDARY MODULE. IT WILL BE CONFIGURED IN THE EXERCISER TO BE THE FIRST MODULE INITIALLY RUN BY THE MONITOR AND SINCE IT IS A "NBKMOOD" IT WILL RUN ONLY ONE PASS AND NEVER RUN AGAIN UNTIL THE EXERCISER IS RESTARTED. ITS PURPOSE IS TO REESTABLISH CONFIDENCE IN THE HARDWARE AFFECTING OVERALL SYSTEM TIMING. IF THE TIMING IS FOUND TO BE WITHIN A PRE-ESTABLISHED EMPIRICALLY DETERMINED LIMIT, THE MODULE WILL ENABLE USE OF THE SYSTEM TIMER MODULE. IF NOT IT WILL REPORT THE ERROR AND DISABLE USE OF THE SYSTEM TIMER MODULE.

2.0 REQUIREMENTS

HARDWARE: ANY PDP11/40 OR 11/34A SYSTEM WITH CORE MEMORY AND A KW11-L LINE CLOCK.

STORAGE:: KWF REQUIRES:

- 1. DECIMAL WORDS: 355
- 2. OCTAL WORDS: 0543
- 3. OCTAL BYTES: 1306

3.0 PASS DEFINITION

ONE PASS OF KWF CONSISTS OF A SINGLE PASS THROUGH THE MODULE CODE TO PERFORM THE REQUIRED DIAGNOSIS AND ANALYSIS.

4.0 EXECUTION TIME

ONE PASS OF KWF RUNNING ON AN 11/40 CPU TAKES APPROXIMATELY ONE MINUTE.

NOTE: THE TIMING INFORMATION IN THE "ENDPAS" MESSAGE FOR THE "KWF" MODULE IS MEANINGLESS SINCE THE TIMER MODULE DOES NOT GET STARTED UNTIL "KWF" REPORTS ITS END OF PASS.

5.0 CONFIGURATION PARAMETERS

DEFAULT PARAMETERS:

DVA: 177546 VCT: 100 BR1: 6 BR2: 0 DVC: 1 SRI: 000037

REQUIRED PARAMETERS:

FOR 50 CYCLE SYSTEMS THE TIMING PARAMETER IN SRI SHOULD BE SET TO 000030. (APPROX)

FOR CPU'S OTHER THAN THE KD11-A, THE VALUE OF SRI WILL HAVE TO BE EMPERICALLY DETERMINED BY RUNNING THIS MODULE ALONE AND NOTING THE INFORMATION PROVIDED IN THE ERROR PRINTOUT.

FOR AN I134A 60 CYCLE SYSTEM THE TIMING PARAMETER IN SRI SHOULD BE SET TO 000031.

6.0 DEVICE OPTION SET-UP  
-----

NONE REQUIRED

7.0 MODULE OPERATION  
-----

A. VERIFY THAT THE KW11-L MONITOR BIT CAN BE SET BY THE POWER SUPPLY SIGNAL. IF A FAILURE IS DETECTED, REPORT THE ERROR AND DROP THE MODULE.

B. VERIFY THAT THE KW11-L CAN GENERATE AN INTERRUPT TO THE PROPER VECTOR WHEN ENABLED. IF A FAILURE IS DETECTED REPORT THE ERROR AND DROP THE MODULE.

C. PERFORM THE TIMING ANALYSIS:

1. SYNC THE LINE CLOCK
2. COUNT THE NO. OF ITERATIONS THROUGH A MONITOR "BREAK" LOOP WHILE WAITING FOR AN INTERRUPT.
3. STORE THE COUNT IN A TABLE
4. REPEAT (1) THRU (3) SIXTEEN TIMES TO STORE 16 COUNTS IN THE TABLE
5. SUM ALL ENTRIES IN THE TABLE AND DIVIDE BY 16 TO GET THE AVERAGE
6. CHECK THAT THE AVERAGE IDS WITHIN LIMITS
7. IF WITHIN TOLERANCE, ENABLE SYSTEM TIMER TO BE ABLE TO RUN AND REPORT END OF PASS.
8. IF NOT WITHIN TOLERANCE, REPORT THE ERROR AND DROP THE MODULE. (SYSTEM TIMER NOT ENABLED TO RUN)

8.0 OPERATOR OPTIONS  
-----

"SRI" AND THE CONTENTS OF LOCATION "TOL" MAY HAVE TO BE MODIFIED FOR UNUSAL CASES TO COMPENSATE FOR A PARTICULAR SYSTEM INSTALLATION. (SRI) IS THE AVERAGE TIMER COUNT AND IF FOUND TO BE 000000, THE MODULE ASSUMES A DEFAULT OF "000037" (KD11-A CPU). IF NOT 000000, THE MODULE TAKES THE CONTENTS OF "SRI" AS THE AVERAGE TIME VALUE. THE LOCATION TAGGED "TOL" IS THE ALLOWABLE TOLERANCE AND IS PROGRAM LOADED AS A "3".

9.0 NON-STANDARD PRINTOUTS  
-----

KWFB DEC/X11 SYSTEM EXERCISER MODULE  
KKWFB0.P11 12-OCT-78 12:07

MACY11 30A(1052) 12-OCT-78 16:48 PAGE 5

SEQ 0004

IF A TIMING ANALYSIS ERROR IS DETECTED IT IS REPORTED  
VIA AN EXTENDED ERROR PRINTOUT OF FOUR OCTAL NUMBERS:

WWWWW SSSSS HHHHH LLLLLL

WHERE:

WWWWW IS THE WAS AVERAGE  
SSSSS IS WHAT THE AVERAGE SHOULD HAVE BEEN  
HHHHH IS THE HIGH LIMIT  
LLLLL IS THE LOW LIMIT

```

000000 N$KMOD <KWFB >,177546,100,6,1,162
000000 MODULE 41000,KWFB,177546,100,6,1,162
          ;TITLE KWFB DEC/X11 SYSTEM EXERCISER MODULE
          ;DDXCOM VERSION 6 23-MAY-78
          ;LIST BIN
*****
000000 053513 041106 040 BEGIN:
000005 000 MODNAM: -ASCII /KWFB / ;MODULE NAME
000006 177546 ADDR: 177546+0 ;USED TO KEEP TRACK OF WBUFF USAGE
000010 000100 VECTOR: 100+0 ;1ST DEVICE ADDR
000012 000 BR1: -BYTE PRTY6+0 ;1ST DEVICE VECT0R.
000013 000 BR2: -BYTE PRTY+0 ;1ST BR LEVEL.
000014 000001 DVID1: +1 ;2ND BR LEVEL.
000016 000000 SR1: OPEN ;DEVICE INDICATOR 1.
000017 000000 SR2: OPEN ;SWITCH REGISTER 1.
000022 000000 SR3: OPEN ;SWITCH REGISTER 2.
000024 000000 SR4: OPEN ;SWITCH REGISTER 3.
          ;SWITCH REGISTER 4
*****
000026 041000 STAT: 41000 ;STATUS WORD.
000030 000224 INIT: START ;MODULE START ADDR.
000032 000224 SPOINT: MODSP ;MODULE STACK POINTER.
000034 000000 PASCNT: 0 ;PASS COUNTER.
000040 000000 ICOUNT: 1 ;NO OF ITERATIONS PER PASS=1
000042 000000 SOFCNT: 0 ;LOC TO COUNT ITERATIONS
000046 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000048 000000 SFPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000050 000000 HRDPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000052 000000 SYSCNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000054 000000 RANUM: 0 ;NO OF SYS ERRORS ACCUMULATED
000056 000000 CONF1G: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000060 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000062 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000064 000000 SVR0: OPEN ;LOC TO SAVE R0.
000066 000000 SVR1: OPEN ;LOC TO SAVE R1.
000070 000000 SVR2: OPEN ;LOC TO SAVE R2.
000074 000000 SVR3: OPEN ;LOC TO SAVE R3.
000076 000000 SVR4: OPEN ;LOC TO SAVE R4.
000078 000000 SVR5: OPEN ;LOC TO SAVE R5.
000080 000000 SVR6: OPEN ;LOC TO SAVE R6.
000100 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
000102 000000 SBADR: OPEN ;ADDR OF GOOD DATA, OR
000104 000000 ACSR: OPEN ;CONTENTS OF CSR.
000106 000000 ASTAT: OPEN ;ADDR OF BAD DATA, OR
000108 000000 ERR1P: OPEN ;STATUS REG CONTENTS.
000110 000000 ASB: OPEN ;TYPE OF ERROR
000112 000322 AMAS: OPEN ;EXPECTED DATA.
000114 000000 RSTRT: RSTRT ;ACTUAL DATA
000116 000000 WDT0: OPEN ;RSTRT ADDRESS AFTER END OF PASS
000120 000000 WDR: OPEN ;WORDS TO MEMORY PER ITERATION
000122 000162 IDNUM: 162 ;WORDS FROM MEMORY PER ITERATION
          ;# OF INTERRUPTS PER ITERATION
          ;MODULE IDENTIFICATION NUMBER=162
          ;REPT SPSIZ ;MODULE STACK STARTS HERE.
          ;NLIST

```

```

          .WORD 0
          .LIST
          .ENDR
000224* MODSP:
*****
232 -GLOBL CLOCK,CLOCKL,CLOCKP,LCLEAR,PCLEAR ;MONITOR AND TIMER MODULE LOCATIONS
233
234
235 000224* 132767 000001 000000G START: BITB #BIT0,CLOCK ;USING A KW11-L TIMER ?
236 000232* 001006 000002 000000G BNE IS ;BR IF YES
237 000242* 001010 000000* BITB #BIT1,CLOCK ;USING THE KW11-P TIMER ?
238 000244* 194410 000000* ENDS,BEGIN ;BR IF YES
239
240 000250* 016700 000000G 1$: MOV CLOCKL,R0 ;GET POINTER TO TIMING INFORMATION
241 000262* 000405 000000G MOV #LCLEAR,TCLEAR ;SET PROPER JSR ADDR
242 BR JS ;CONTINUE
243 000270* 012767 000000G 2$: MOV CLOCKP,R0 ;GET POINTER TO TIMING INFORMATION
244 000276* 005720 000000G MOV #PCLEAR,TCLEAR ;SET PROPER JSR ADDR
245 BR JS ;CONTINUE
246 000300* 005010 000000G 3$: TST (R0)+ ;POINT TO LOCATION CONTAINING "TIME"
247 000302* 004777 000710 CLR (R0) ;INIT RUN TIME = 000000
248 000306* 005767 177504 PC,#ATCLEAR ;CLEAR MODULE PASSTIME TABLE IN CLOCK MODULE
249 000312* 001003 000037 177474 BNE RSTRT ;DOES SR1 SPECIFY A TIME ?
250 000314* 012767 000037 177474 MOV #37,SR1 ;BR IF IT DOES
251 ;DEFAULT TO 37(9) - KD11-A CPU
252 000322* 016767 177460 RSTRT: MOV ADDR,CSRA ;SAVE KW11-L ADDRESS
253 000330* 016760 177454 MOV VECTOR,R0 ;GET VECTOR ADDRESS
254 000334* 012720 001124* MOV #KWINT,(R0)+ ;STEER KW INTR'S TO KWINT
255 000340* 116710 177446 MOV #R1,(R0)
256 000344* 005077 177436 CLR @ADDR
257
258 ;VERIFY THAT KW MONITOR BIT (BIT07) CAN BE SET BY POWER SUPPLY SIGNAL
259 -----
260 000350* 005005
261 000352*
262 000352* 104407 000000* KWDT1: CLR R5 ;INIT TIMER
263 000352* 104407 000000* 1$: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
264 000356* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
265 000362* 005305 DEC R5 ;COUNT TIMER
266 000364* 005305 BNE #200,@ADDR ;BR UNTIL TIMER = 0
267 000366* 022777 000200 177412 BEQ KWDT2 ;DID MONITOR BIT SET?
268 000374* 001415 MOV @ADDR,ACSR ;SAVE CONTENTS OF LKCSR
269 000376* 011415 MOV #25,ERRTYD
270 000404* 012767 000025 177474 *****
271 *****
272 000412* 104405 000000* 000000 HDRERS,BEGIN,NULL ;KW11-L MONITOR BIT WON'T SET
273 *****
274 000420* 105067 000000G CLR# CLOCK ;DISABLE ALL TIMING CHECKS
275 000424* 104410 000000G ENDS,BEGIN
276
277 ;VERIFY THAT KW11-L CAN GENERATE AN INTERRUPT PROPERLY
278 -----
279
280 000430* 005067 000566 KWDT2: CLR #KWFLG1 ;CLEAR SOFTWARE INTR. FLAG
281 000434* 005005 CLR R5 ;INIT TIMER
282 000436* 052777 000100 177342 BIS #100,@ADDR ;SET LKCSR INTR ENAB

```

283 000444 042777 000200 177334  
284 000452 005767 000544  
285 000452 104410 000000  
286 000460 104407 000000  
287 000464 104407 000000  
288 000470 005305  
289 000470 013669  
290 000474 017767 177306 177400  
291 000502 005077 177300  
292 000506 012767 000023 177372  
293  
294 000514 104405 000000 000000  
295  
296 000522 105067 000000G  
297 000526 104410 000000  
298 000532 032777 000100 177246  
299 000540 001417  
300 000546 017767 177240 177332  
301 000550 005077 177322  
302 000554 012767 000023 177324  
303  
304 000562 104405 000000 000000  
305  
306 000570 105067 000000G  
307 000574 104410 000000G  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338

```
1S: BIC #200,@ADDR ;CLEAR MONITOR BIT
TST KWFLG1 ;DID KW11-L INTR OCCUR
BMT 2S ;BR IF YES
BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
DEC R5 ;COUNT TIMER
BNE 1S ;BR IF NO TIMEOUT
MOV @ADDR,ACSR ;SAVE CONTENTS OF LKCSR
CLR @ADDR ;CLEAR INTR ENAB
MOV #23,ERRTYP
;*****
HRDRS,BEGIN,NULL ;KW11-L FAILED TO INTR ON TIME
;*****
CLRB CLOCK ;DISABLE ALL TIMING CHECKS
ENDS,BEGIN
2S: BIT #100,@ADDR ;DID INTR SERVICE CLEAR BIT 06?
BEQ KWFLG1 ;BR IF YES
MOV @ADDR,ACSR ;SAVE CONTENTS OF LKCSR
CLR @ADDR ;CLEAR LKCSR
MOV #25,ERRTYP
;*****
HRDRS,BEGIN,NULL ;KW11 INTR SERVICE FAILED TO CLR BIT06
;*****
CLRB CLOCK ;DISABLE ALL TIMING CHECKS
ENDS,BEGIN
;GROSS TIMING ANALYSIS ROUTINE - PERFORMS GROSS CPU/MEMORY TIMING ANALYSIS
;USING THE KW11-L
;*****
KWDT3: JSR PC,CLRTAB ;GO CLEAR LOOP COUNTERS TABLE
MOV #CLRTAB,R1 ;R1 POINTS TO LOOP COUNTERS TABLE
KWSYNC: CLR KWFLG1 ;INIT SOFTWARE INTR. FLAG
CLR R0 ;INIT TIMER
CLR #100,@ADDR ;ENABLE INTRS
TST KWFLG1 ;INTR OCCUR YET?
BMT 2S ;BR IF YES
BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
DEC R0 ;COUNT TIMER
BNE 1S ;BR IF NO TIMEOUT
MOV @ADDR,ACSR ;SAVE CONTENTS OF LKCSR
CLR @ADDR ;CLEAR OUT LKCSR
MOV #23,ERRTYP
;*****
HRDRS,BEGIN,NULL ;KW11-L TIMEOUT
;*****
CLRB CLOCK ;DISABLE ALL TIMING CHECKS
ENDS,BEGIN
3S: CLR KWFLG1 ;INIT SOFTWARE INTR FLAG
CLR R0 ;INIT LOOP COUNTER
BIS #100,@ADDR ;ENABLE KW11-L INTRS
TST KWFLG1 ;INTR OCCUR YET?
BMT 4S ;BR IF YES
BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
```

339 000732 104407 000000  
340 000736 005200  
341 000740 001367  
342 000742 104407 177040 177132  
343 000750 005077 177032  
344 000754 012767 000023 177124  
345  
346 000762 104405 000000 000000  
347  
348 000770 105067 000000G  
349 000774 104410 000000  
350 001000 010021  
351 001002 022701 001306  
352 001006 001300  
353 001010 004767 000140  
354 001014 010003  
355 001016 016701 176774  
356 001022 010102  
357 001024 066701 000170  
358 001030 166702 000164  
359 001034 160301  
360 001036 103406  
361 001040 160203  
362 001042 103404  
363 001044 104413 000000  
364  
365 001050 000167 177246  
366 001054 105067 000000G  
367 001060 060001  
368 001062 012704 001236  
369 001066 010024  
370 001070 016724 176722  
371 001074 010174  
372 001076 010224  
373 001100 017767 176702 176774  
374 001106 005067 176774  
375  
376 001112 104405 000000 001224  
377  
378 001120 104410 000000  
379  
380  
381  
382  
383  
384  
385  
386 001124 005077 176656  
387 001130 005167 000066  
388 001134 000002  
389  
390  
391  
392  
393 001136 012701 001246  
394 001142 005021

```
3S: BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
INC R0 ;COUNT ONE PASS THRU BREAK LOOP
BNE 3S ;BR IF NO TIMEOUT
MOV @ADDR,ACSR ;SAVE CONTENTS OF LKCSR
CLR @ADDR ;CLEAR OUT LKCSR
MOV #23,ERRTYP
;*****
HRDRS,BEGIN,NULL ;KW11-L TIMEOUT
;*****
CLRB CLOCK ;DISABLE ALL TIMING CHECKS
ENDS,BEGIN
4S: MOV R0,(R1)+ ;SAVE LOOP COUNT IN TABLE
CND #CLRTAB+4,R1 ;END OF TABLE ??
BNE KWSYNC ;GO DO IT AGAIN
JSR PC,TIMAVG ;GO COMPUTE TABLE AVERAGE
MOV R0,R3 ;R3 CONTAINS AVERAGE COUNT
MOV R1,R2 ;GET WHAT AVERAGE COUNT SHOULD BE
PUT IT IN R2 TOO
ADD TOL,R1 ;R1 CONTAINS HIGH LIMIT COUNT
SUB TOL,R2 ;R2 CONTAINS LOW LIMIT COUNT
BCS R3,R1 ;LESS THAN HIGH LIMIT?
SUB R2,R3 ;GREATER THAN LOW LIMIT?
BNC 4S ;BR IF NOT
BCS SS ;MONITOR SHALL TEST END OF PASS
ENDITS,BEGIN
5S: JMP RESTRT ;DISABLE USING THE TIMER MODULE
CLRB CLOCK ;RESTORE HIGH LIMIT
ADD R0,R1 ;POINT TO SAVE AREA
MOV #TIMES,R4 ;SAVE WAS COUNTER AVERAGE
MOV R0,(R4)+ ;SAVE SHOULD BE COUNTER AVERAGE
MOV R1,(R4)+ ;SAVE HIGH LIMIT
MOV R2,(R4)+ ;SAVE LOW LIMIT
MOV @ADDR,ACSR ;SAVE CONTENTS OF LKCSR
CLR ERRTYP
;*****
HRDRS,BEGIN,KWMSG ;PRINT COUNTER INFORMATION
;*****
ENDS,BEGIN
;GENERAL PURPOSE UTILITY ROUTINES
;*****
;*****
;INTERRUPT SERVICE ROUTINE
;*****
KWINT: CLR @ADDR ;CLEAR OUT LKCSR
COM KWFLG1 ;SET SOFTWARE INTR. FLAG
RTI ;RETURN TO MAINLINE
;ROUTINE TO CLEAR LOOP COUNTER TABLE
;*****
CLRTAB: MOV #CLRTAB,R1 ;POINT TO FIRST TABLE ENTRY
1S: CLR (R1)+ ;CLEAR ONE WORD
```





