

.REM_

IDENTIFICATION

PRODUCT CODE: AC-E920B-MC
PRODUCT NAME: CXKWEB0 KVV11K 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

1.0 ABSTRACT

THE KWE IS AN IOMOD THAT EXERCISES THE KVV11K REAL TIME CLOCK. ON START IT EXERCISES THE CSRS AND PRESET BUFFERS OF THE CLOCK, THEN AND FOR ALL RESTARTS, IT EXERCISES THE CLOCK AT EACH ONE OF ITS BASIC RATES.

2.0 REQUIREMENTS

HARDWARE: ONE KVV11

STORAGE:: KWE REQUIRES:

1. DECIMAL WORDS: 502
2. OCTAL WORDS: 0766
3. OCTAL BYTES: 1754

3.0 PASS DEFINITION

ONE PASS OF THE KWE MODULE CONSISTS OF GENERATING INTERRUPTS FOR ONE SECOND AT EACH CLOCK RATE, UNTIL 60 SECONDS HAVE ELAPSED.

4.0 EXECUTION TIME

ONE PASS OF THE KWE MODULE RUNNING ALONE TAKES APPROXIMATELY ONE MINUTE.

5.0 CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 170420, VECTOR 440, BR1: 4

DEVcnt: 1, SR1: 0

REQUIRED PARAMETERS:

NONE

6.0 DEVICE/OUTPUT SET-UP:

NONE

7.0 MODULE OPERATION

TEST SEQUENCE:

1. (START) BIT EXERCISE CSR,PRESET REGISTER OF CLOCK.
2. (RESTR) COUNT TESTS USING INTERRUPTS COUNT INTERRUPTS WILL OCCUR IN ONE SECOND AND ADVANCE THE TEST TO THE NEXT RATE.

AFTER A RATE HAS BEEN SELECTED, A CHECK IS MADE TO SEE IF THE OPERATOR HAS INHIBITED THAT RATE FROM TEST. IF NOT, CONTROL IS TRANSFERRED TO THE PARTICULAR RATE ROUTINE (LISTED BELOW). EACH RATE ROUTINE MUST PRELOAD THE BUFFER REGISTER OF THE CLOCK TO THE COUNT THAT WILL CAUSE IT TO INTERRUPT IN ONE SECOND. AFTER THE BUFFER IS LOADED, THE CSR IS LOADED WITH THE PROPER BITS THAT SELECT THE RATE.

- A. COUNT TEST CLOCK A RATE 1MHZ.
- B. CLOCK A RATE: 100KHZ
- C. CLOCK A RATE: 10KHZ
- D. CLOCK A RATE: 1KHZ
- E. CLOCK A RATE: 100HZ
- F. CLOCK A RATE: LINE FREQ.
- G. CLOCK A RATE: PSEUDO RANDOM (1 OF 3 RATES)

8.0 OPERATION OPTIONS

VALID SR1 VALUES

SR1 BIT	ENABLE/DISABLE	FUNCTION
0	0	ENABLE TESTING 1MHZ
	1	DISABLE TESTING 1MHZ
1	0	ENABLE TESTING 100KHZ
	1	DISABLE TESTING 100KHZ
2	0	ENABLE TESTING 10KHZ
	1	DISABLE TESTING 10KHZ
3	0	ENABLE TESTING 1KHZ
	1	DISABLE TESTING 1KHZ
4	0	ENABLE TESTING 100HZ
	1	DISABLE TESTING 100HZ
5	0	*ENABLE TESTING RANDOM
	1	DISABLE TESTING RANDOM
6	0	ENABLE TESTING LINE FREQ
	1	DISABLE TESTING LINE FREQ

*NOTE: IF RANDOM RATE SELECTED, THEN AN SR1 BIT DISABLING A PARTICULAR RATE WILL BE IGNORED.

9.0 NON-STANDARD PRINTOUTS:

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

```
177 000000- IOMOD <KWEB>,170420,440,6,6,60,122
178 000000- MODULE 140000,KWEB,170420,440,6,6,60,122
179 .TITLE KWEB DEC/X11 SYSTEM EXERCISER MODULE
180 ; DDXCOM VERSION 6 23-MAY-78
181 ; LIST BIN
182 ;*****
183 BEGIN:
184 MODNAM: .ASCII /KWEB / ;MODULE NAME
185 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
186 ADDR: 170420+0 ;1ST DEVICE ADDR
187 VECTOR: 440+0 ;1ST DEVICE VECTOR
188 BR1: .BYTE PRTY6+0 ;1ST BR LEVEL
189 BR2: .BYTE PRTY6+0 ;2ND BR LEVEL
190 DVID1: +1 ;DEVICE INDICATOR 1
191 SR1: OPEN ;SWITCH REGISTER 1
192 SR2: OPEN ;SWITCH REGISTER 2
193 SR3: OPEN ;SWITCH REGISTER 3
194 SR4: OPEN ;SWITCH REGISTER 4
195 ;*****
196 STAT: 140000 ;STATUS WORD
197 INIT: START ;MODULE START ADDR
198 SPOINT: MODSP ;MODULE STACK POINTER
199 PASCNT: 0 ;PASS COUNTER
200 ICNT: 60. ;# OF ITERATIONS PER PASS=60.
201 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
202 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
203 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
204 SOPPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
205 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
206 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
207 RANNUM: 0 ;HCLDS RANDOM # WHEN RAND MACRO IS CALLED
208 CONFIG: ;RESERVED FOR MONITOR USE
209 RES1: 0 ;RESERVED FOR MONITOR USE
210 RES2: 0 ;RESERVED FOR MONITOR USE
211 SVR0: OPEN ;LOC TO SAVE R0.
212 SVR1: OPEN ;LOC TO SAVE R1.
213 SVR2: OPEN ;LOC TO SAVE R2.
214 SVR3: OPEN ;LOC TO SAVE R3.
215 SVR4: OPEN ;LOC TO SAVE R4.
216 SVR5: OPEN ;LOC TO SAVE R5.
217 SVR6: OPEN ;LOC TO SAVE R6.
218 CSRA: OPEN ;ADDR OF CURRENT CSR.
219 SBADR: ;ADDR OF GOOD DATA, OR
220 ACSR: OPEN ;CONTENTS OF CSR.
221 BASADR: ;ADDR OF BAD DATA, OR
222 ASAT: OPEN ;STATUS REG CONTENTS.
223 ERRTYP: ;TYPE OF ERROR
224 ASB: OPEN ;EXPECTED DATA.
225 AWAS: OPEN ;ACTUAL DATA.
226 RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
227 WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
228 WDPR: OPEN ;WORDS FROM MEMORY PER ITERATION
229 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
230 IDNUM: 122 ;MODULE IDENTIFICATION NUMBER=122
231 .REPT SPSIZ ;MODULE STACK STARTS HERE.
232 .NLIST
```

```
233 .WORD 0
234 .LIST
235 .ENDR
236 000224-
237 MODSP:
;*****
```

```
238 ;MODULE REQUIRED REGISTERS - SET UP BY THIS MODULE.
239
240 ASR: .WORD 170420 ;CLOCK A STATUS REG.
241 ABR: .WORD 170422 ;CLOCK A BUFFER REG.
242
243
244 AVECT: .WORD 440 ;CLOCK A INTERRUPT VECTOR.
245 AVECT2: .WORD 442
246
247 BVECT: .WORD 444 ;CLOCK INTERRUPT VECTOR.
248 BVECT2: .WORD 446
249
250
251 RATEP: .WORD 1 ;POINTS TO CURRENT RATE
252 OFF: .WORD 0 ;OFFSET TO TAKE US TO RATE ROUTINE
253 RAW: .WORD 0 ;RANDOM NUMBER.
254 RANB: .WORD 0 ;RANDOM NUMBER.
255 AIFLG: .WORD 0 ;FLAG TO SHOW THAT CLOCK A HAS INTERRUPTED.
256 TRY: 0 ;*****
257
258 START:
259 MOV #7,INTR ;7 INTERRUPTS/ITERATION
260
261 MOV ADDR,ASR ;GET BASE ADDR.
262 MOV VECTOR,AVECT ;GET BASE VECTOR ADDR.
263
264 MOV ASR,ABR ;NOW WE'RE GONNA FIX
265 ADD #2,ABR ;ALL CLOCK ADDRESSES BASED ON ASR.
266
267 MOV AVECT,R0 ;NOW FIX VECTOR ADDRESSES
268 ADD #4,R0
269 MOV AVECT,AVECT
270 MOV AVECT,AVECT2
271 ADD #2,AVECT2
272 MOV BVECT,BVECT2
273 ADD #2,BVECT2
274
275 ;*
276 ;*LOGIC TEST #1 BE SURE A CLOCK EXISTS AT THE
277 ;*SPECIFIED ADDR IF NO CLOCK THEN A
278 ;*DEC/X11 SYS ERROR WILL OCCUR.
279 ;*
280 LOG1: TST @ASR ;ADDRESS THE CLOCK. IF SYS ERROR
281 ;OCCURS, THEN CLOCK DID NOT
282 ;RETURN SLAVE-SVN WHEN
283 ;ADDRESSED.
284
285 ;*
286 ;*LOGIC TEST #2. MAKE SURE CLOCK CSR BITS
287 ;*14,11,6,5,2, AND 0 CAN BE SET + CLEARED.
288 ;*
289 LOG2: MOV #044125,ASTAT ;GENERATE + RECORD PATTERN TO BE USED.
290 MOV ASTAT,@ASR ;SET THEM IN CSR OF CLOCK A.
291 MOV @ASR,ACSR ;READ THEM BACK
292
293
```

```
294 CMP ASTAT,ACSR ;DID THEY ALL SET?
295 BEQ 25 ;YES - GO TO NEXT TEST.
296 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.
297 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
298
299 1$: MOV ASR,CSRA ;RECORD CSR'S ADDR
300 ADD #25,ERRTYP ;BIT STUCK
301 ;*****
302 ;*****
303 ;*****
304
305 2$: CLR @ASR ;TRY CLEARING THE BITS
306 MOV @ASR,ACSR ;READ IT BACK.
307 BEQ LOG ;IF ZERO CSR GOOD.
308 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.
309 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
310
311 3$: CLR ASTAT ;EXPECT ZERO CSR
312 MOV ASR,CSRA ;RECORD CSR'S ADDR.
313 ADD #25,ERRTYP ;BIT STUCK
314 ;*****
315 ;*****
316 ;*****
317 ;*
318 ;*LOGIC TEST #3. MAKE SURE CLOCK CSR BITS
319 ;*13,5,3, AND 1 CAN BE SET + CLEARED.
320 ;*
321 LOG3: MOV #020052,ASTAT ;GENERATE + RECORD PATTERN TO BE USED.
322 MOV ASTAT,@ASR ;SET THEM IN CSR OF CLOCK A.
323 MOV @ASR,ACSR ;READ THEM BACK
324 CMP ASTAT,ACSR ;DID THEY ALL SET?
325 BEQ 25 ;YES - GO TO NEXT TEST.
326 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.
327 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
328
329 1$: MOV ASR,CSRA ;RECORD CSR'S ADDR.
330 ADD #25,ERRTYP ;BIT STUCK
331 ;*****
332 ;*****
333 ;*****
334
335 2$: CLR @ASR ;TRY CLEARING THE BITS
336 MOV @ASR,ACSR ;READ IT BACK.
337 BEQ LOG ;IF ZERO CSR GOOD.
338 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.
339 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
340
341 3$: CLR ASTAT ;EXPECT ZERO CSR.
342 MOV ASR,CSRA ;RECORD CSR'S ADDR.
343 ADD #25,ERRTYP ;BIT STUCK
344 ;*****
345 ;*****
346 ;*****
347 ;*
348 ;*LOGIC TEST #4. MAKE SURE CLOCK BUFFER REG
349
```

```

350 ;*PATTERN 125252 CAN BE SET + CLEARED.
351 ;*
352
353 000656 012767 125252 177220 LOG4: MOV #125252,ASTAT ;GENERATE + RECORD PATTERN TO BE USED.
354 000664 016777 177214 177334 MOV ASTAT,@ABR ;SET THEM IN BUFFER REG OF CLOCK.
355 000672 017767 177330 177202 MOV @ABR,ACSR ;READ THEM BACK
356 000700 026767 177200 177174 CMP ASTAT,ACSR ;DID THEY ALL SET?
357 000706 031415 BQ ;YES - GO TO NEXT TEST.
358 000710 104407 000000 BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR....
359 000710 104407 000000 BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
360
361 000720 016767 177302 177152 1S: MOV ABR,CSRA ;RECORD BUFFER REG ADDR.
362 000726 012767 000025 177152 MOV #25,ERRTYP ;BIT STUCK
363 ***** ;*****
364 000734 104405 000000 000000 HRDR$,BEGIN,NULL ;BUFFER REG PATTERN 125252 FAILED
365 ***** ;*****
366
367 000742 005077 177260 2S: CLR @ABR ;TRY CLEARING THE BITS
368 000746 017767 177254 177126 MOV @ABR,ACSR ;READ IT BACK.
369 000754 001417 BQ ;IF ZERO BUFFER GOOD.
370 000756 104407 000000 BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR....
371 000762 104407 000000 BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
372 000766 005067 177112 3S: CLR ASTAT ;EXPECT ZERO BUFFER.
373 000770 016767 177130 MOV ABR,CSRA ;RECORD ADDR. OF BUFFER REG.
374 001000 012767 000025 177100 MOV #25,ERRTYP ;BIT STUCK
375 ***** ;*****
376 001006 104405 000000 000000 HRDR$,BEGIN,NULL ;BUFFER REG FAILED TO CLEAR
377 ***** ;*****
378
379 ;*
380 ;*LOGIC TEST #5 MAKE SURE CLOCK BUFFER REG
381 ;*PATTERN 052525 CAN BE SET + CLEARED.
382 ;*
383
384 001014 012767 052525 177062 LOG5: MOV #052525,ASTAT ;GENERATE + RECORD PATTERN TO RE USED.
385 001022 016777 177056 177176 MOV ASTAT,@ABR ;SET THEM IN BUFFER OF CLOCK A.
386 001030 017767 177172 177044 MOV @ABR,ACSR ;READ THEM BACK
387 001036 026767 177042 177036 CMP ASTAT,ACSR ;DID THEY ALL SET?
388 001044 001415 BQ ;YES - GO TO NEXT TEST.
389 001048 104407 000000 BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR....
390 001054 104407 000000 BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
391
392 001056 016767 177144 177014 1S: MOV ABR,CSRA ;RECORD BUFFER REG ADDR.
393 001064 012767 000025 177014 MOV #25,ERRTYP ;BIT STUCK
394 ***** ;*****
395 001072 104405 000000 000000 HRDR$,BEGIN,NULL ;BUFF REG PATTERN 052525 FAILED
396 ***** ;*****
397
398 001100 005077 177122 2S: CLR @ABR ;TRY CLEARING THE BITS
399 001104 017767 177116 176770 MOV @ABR,ACSR ;READ IT BACK.
400 001112 001417 BQ ;IF ZERO BUFFER GOOD.
401 001114 104407 000000 BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR....
402 001120 104407 000000 BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
403 001124 005067 176754 3S: CLR ASTAT ;EXPECT ZERO BUFFER.
404 001130 016767 177072 176742 MOV ABR,CSRA ;RECORD BUFFER REG A ADDR.
405 001136 012767 000025 176742 MOV #25,ERRTYP ;BIT STUCK

```

```

406 ***** ;*****
407 001144 104405 000000 000000 HRDR$,BEGIN,NULL ;BUFFER REG FAILED TO CLEAR
408 ***** ;*****
409
410 001152 012767 125252 177064 RESTR: MOV #125252,RANA ;PRESET RANDOM NUMBER GENERATOR
411 001160 013767 000100 177052 MOV #BIT6,RATEP ;FIRST PASS THRU LOOP, OFFSET=0, RATES WILL=1.
412 001166 012777 001626 177034 MOV #INSEV,RAVECT ;SET UP CLOCK A'S INTER. VECTOR.
413 001174 116777 176612 177030 MOV#R1,RAVECT2 ;SET PRIORITY ON CLOCK A'S INTR.
414 001202 106167 177032 LOOP: ROLB RATEP ;GET NEXT RATE.
415
416 BPL IS ;IF NOT END THEN CONTINUE.
417 CLR OFF ;CLEAR THE OFFSET
418 MOV #1,RATEP ;LOOK AT FIRST RATE
419 ADD #2,OFF ;
420 BIT RATEP,SRI ;IS THIS RATE INHIBITED?
421 RNE LOOP ;
422
423 001240 005067 177004 CLR AIFLG ;CLR FLAG INDICATING CLOCK A HAS INTERRUPTED.
424 001244 016701 176772 MOV OFF,R1 ;PICK UP OFFSET
425 001250 000171 001254 JMP @LISTP(R1) ;GO SET THE RATE + START THE CLOCK.
426
427 ;
428 ;THE FOLLOWING (LISTP) ARE POINTERS TO VARIOUS RATE
429 ;ROUTINES. THEY ARE USED BY "LOOP". "LOOP" GENERATES
430 ;AN OFFSET OF A RATE WE WISH TO EXERCISE. THE OFFSET
431 ;IS STORED IN R1. WE INDEX "LISTP" BY R1 (JMP @LISTP(R1))
432 ;TO GET THE ADDRESS OF THE RATE ROUTINE TO EXERCISE.
433 ;
434 LISTP: .WORD 1 ;
435 .WORD RATE0 ;POINTER TO 1MHZ ROUTINE
436 .WORD RATE1 ;POINTER TO 100KHZ ROUTINE
437 .WORD RATE2 ;POINTER TO 10KHZ ROUTINE
438 .WORD RATE3 ;POINTER TO 1KHZ ROUTINE
439 .WORD RATE4 ;POINTER TO 100HZ ROUTINE
440 .WORD RATE5 ;POINTER TO RANDOM ROUTINE
441 .WORD RATE6 ;POINTER TO LINE FREQ. ROUTINE
442
443 ;
444 ;THE FOLLOWING (RATEAL) ARE THE PRESET VALUES THAT THE
445 ;VARIOUS RATE ROUTINES NEED. THEY ARE LOADED INTO
446 ;CLOCK A'S PRESET BUFFER. "RATEAL" IS INDEXED BY
447 ;AN OFFSET IN R1 BY THE RATE ROUTINES TO GET THE
448 ;PRESET VALUE
449 ;
450 RATEAL: .WORD 1 ;OFFSET ZERO, NO RATE.
451 .WORD -50000. ;VALUE FOR 1MHZ PRESET.
452 .WORD -50000. ;PRESET VALUE FOR 100 KHZ
453 .WORD -1000. ;PRESET VALUE FOR 10 KHZ
454 .WORD -1000. ;PRESET VALUE FOR 1 KHZ
455 .WORD -100. ;PRESET VALUE FOR 100 HZ
456 .WORD 0 ;PRESET VALUE FOR RANDOM
457 .WORD -60. ;PRESET VALUE FOR LINE FREQ.
458
459
460
461

```

```

462 ;THE FOLLOWING (RSAL) IS USED BY THE RANDOM
463 ;RATE ROUTINE (RATES). THEY ARE THE VALUES NEEDED
464 ;TO BE PUT INTO THE CLOCK'S CSR FOR A PARTICULAR RATE.
465
466
467 001314* 000000 RSAL: -WORD 0 ;OFFSET ZERO, NO RATE.
468 001316* 000113 -WORD 113 ;1 MHZ, GO., MODE 1
469 001320* 000123 -WORD 123 ;100 KHZ, GO., MODE 1
470 001322* 000131 -WORD 131 ;10 KHZ, GO.
471 001324* 000141 -WORD 141 ;1 KHZ, GO.
472 001326* 000151 -WORD 151 ;100 HZ, GO.
473 001330* 000171 -WORD 171 ;LINE FREQ.
474
475 ;*
476 ;*THIS ROUTINE PRESETS CLOCK A FOR
477 ;*1 MHZ RATE CLOCK A INTRs IN 1/20 SEC., 25 TIMES.
478 ;*
479 ;*
480
481 001332* RATE0: CLR @ASR ;CLEAR CLOCK A
482 001332* 005077 176666 MOV RATEAL(R1),@ABR ;PRESET COUNT IN CLOCK A.
483 001336* 016177 001274* 176662 MOV RSAL(R1),@ASR ;START CLOCK A
484 001344* 016177 001314* 176652 ;NOW WAIT FOR INTERRUPT.
485 ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
486 001352* 104400 000000* EXITS,BEGIN
487 ;*
488 ;*THIS ROUTINE PRESETS CLOCK A FOR
489 ;*100 KHZ RATE CLOCK A INTRs IN .5 SEC., TWICE.
490 ;*
491 ;*
492
493 001356* RATE1: CLR @ASR ;CLEAR CLOCK A
494 001356* 005077 176642 MOV RATEAL(R1),@ABR ;PRESET COUNT IN CLOCK A.
495 001362* 016177 001274* 176636 MOV RSAL(R1),@ASR ;START CLOCK A
496 001370* 016177 001314* 176626 ;NOW WAIT FOR INTERRUPT.
497 ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
498 001376* 104400 000000* EXITS,BEGIN
499 ;*
500 ;*THIS ROUTINE PRESETS CLOCK A FOR
501 ;*10 KHZ RATE CLOCK A INTRs IN 1.0 SEC.
502 ;*
503 ;*
504
505 001402* RATE2: CLR @ASR ;CLEAR CLOCK A
506 001402* 005077 176616 MOV RATEAL(R1),@ABR ;PRESET COUNT IN CLOCK A.
507 001406* 016177 001274* 176612 MOV RSAL(R1),@ASR ;START CLOCK A
508 001414* 016177 001314* 176602 ;NOW WAIT FOR INTERRUPT.
509 ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
510 001422* 104400 000000* EXITS,BEGIN
511 ;*
512 ;*THIS ROUTINE PRESETS CLOCK FOR
513 ;*1 KHZ RATE CLOCK A INTRs IN 1.0 SEC.
514 ;*
515 ;*
516

```

```

518 ;*
519
520 001426* RATE3: CLR @ASR ;CLEAR CLOCK A
521 001426* 005077 176572 MOV RATEAL(R1),@ABR ;PRESET COUNT IN CLOCK A.
522 001432* 016177 001274* 176566 MOV RSAL(R1),@ASR ;START CLOCK A
523 001440* 016177 001314* 176556 ;NOW WAIT FOR INTERRUPT.
524 ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
525 001446* 104400 000000* EXITS,BEGIN
526 ;*
527 ;*THIS ROUTINE PRESETS CLOCK FOR
528 ;*100 HZ RATE CLOCK A INTRs IN 1.0 SEC.
529 ;*
530 ;*
531
532 001452* RATE4: CLR @ASR ;CLEAR CLOCK A
533 001452* 005077 176546 MOV RATEAL(R1),@ABR ;PRESET COUNT IN CLOCK A.
534 001458* 016177 001274* 176542 MOV RSAL(R1),@ASR ;START CLOCK A
535 001464* 016177 001314* 176532 ;NOW WAIT FOR INTERRUPT.
536 ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
537 001472* 104400 000000* EXITS,BEGIN
538 ;*
539 ;*THIS ROUTINE PRESETS CLOCK FOR
540 ;*RANDOM RATES
541 ;*RANDOM RATES
542
543 001476* RATE5: JSR PC,RANDOM ;GET 2 RANDOM NUMBERS.
544 ;
545 001502* RIC #177771,RANA ;MAKE NUMBER < 10.
546 001510* BIC #177771,RANB ;MAKE 2ND NUMBER < 10.
547 ;
548 ;NUMBERS MUST BE 2, 4, OR 6
549 3$: TST RANA ;IS NUMBER ZERO?
550 BNE 4$ ;NO-GO AHEAD.
551 ADD #2,RANA ;MAKE IT NON-ZERO
552
553 4$: TST RANB ;IS NUMBER ZERO?
554 BNE 5$ ;NO-GO AHEAD.
555 ADD #2,RANB ;MAKE IT NON-ZERO.
556
557 5$:
558
559 001546* CLR @ASR ;CLEAR CLOCK A
560 001546* 005077 176452 MOV RANA,R1 ;RECORD THE OFFSET.
561 001556* 010167 176460 MOV R1,OFF ;PRESET CLOCK A.
562 001562* 016177 001274* 176436 MOV RATEAL(R1),@ABR ;START CLOCK A.
563 001570* 016177 001314* 176426 ;NOW WAIT FOR INTERRUPT.
564 ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
565 001576* 104400 000000* EXITS,BEGIN
566 ;*
567 ;*THIS ROUTINE PRESETS CLOCK FOR
568 ;*LINE FREQ RATE CLOCK A INTRs IN 1.0 SEC
569 ;*
570 ;*
571
572 001602* RATE6: CLR @ASR ;CLEAR CLOCK A.
573 001606* 016177 001274* 176412 MOV RATEAL(R1),@ABR ;PRESET COUNT IN A.

```


ICOUNT	000036R	200#																		
ICOUNT	000040R	201#																		
IDNUM	000122R	230#																		
INE	001116R	607#	605#																	
INT	000030R	197#																		
INSERV	001626R	412#																		
INTR	000120R	229#																		
L1STP	001354R	265#																		
LOG1	000354R	265#																		
LOG2	000362R	291#																		
LOG3	000520R	307#																		
LOG4	000656R	338#																		
LOG5	001014R	369#																		
LOOP	001202R	414#																		
MAP22S	104416	238#																		
MODNAM	000000R	184#																		
MODSP	000224R	184#																		
MSCHS	104403	238#																		
MSCSS	104402	238#																		
MSCV	104401	238#																		
OFF	000000	238#																		
OPEN	000242R	252#	302	314	333	345	364	376	395	407										
	000000	185#	417*	419*	424	560*	587	592												
		185#	191	192	193	194	211	212	213	214	215	216	217	218						
		222#	224	225	225	227	228	229	233#											
OTDAS	104420	258#																		
PASCNT	000034R	199#																		
PIRGS	000004	238#																		
POPSP	000720	238#																		
POPSP2	000720	238#																		
PRTY	000000	238#																		
PRTY0	000000	238#																		
PRTY1	000040	238#																		
PRTY2	000100	238#																		
PRTY3	000140	238#																		
PRTY4	000200	238#																		
PRTY5	000400	238#																		
PRTY6	000500	188#																		
PRTY7	000340	238#	189	238#																
PS	000340	238#																		
PSM	177776	238#																		
PUSH	005746	238#																		
PUSH2	024646	238#																		
RANA	000244R	253#	410*	545*	549	551*	559	619	620*	621*										
RANS	000746R	253#	553	555*	555*	619*	621	622*												
RANDOM	001720R	543#	619#																	
RANDS	104417	238#																		
RANNUM	000054R	207#																		
RATEAL	001774R	207#	483	496	509	522	535	561	573											
RATEBL	000540R	251#	414*	414*	418*	426														
RATEC	000112R	226#																		
RATE1	001356R	437#																		
RATE2	001407R	437#																		
RATE3	001426R	439#																		
RATE4	001452R	440#																		
RATE5	001476R	441#																		
RATE6	001602R	442#																		

RESTR	001152R	226#	400	410#																
RES1	000060R	210#																		
RES2	000060R	210#																		
RSAL	001314R	467#	484	497	510	523	536	562	574											
RSTR	000112R	226#																		
SBADR	000102R	219#																		
SOPCNT	000042R	202#																		
SOPERS	104406	238#																		
SOPPAS	000046R	204#																		
SPLIT	000032R	198#																		
SPSIZ	000040	1#																		
SR1	000016R	191#																		
SR2	000020R	192#																		
SR3	000022R	193#																		
SR4	000024R	194#																		
START	000254R	197#																		
STAT	000026R	196#																		
SVR0	000062R	211#																		
SVR1	000064R	212#																		
SVR2	000066R	213#																		
SVR3	000070R	214#																		
SVR4	000072R	215#																		
SVR5	000074R	216#																		
SVR6	000076R	217#																		
SVSCNT	000052R	206#																		
TRPDFD	000022	238#																		
TRY	000252R	238#																		
VECTOR	000010R	187#																		
WASADR	000104R	221#																		
WDFR	000116R	220#																		
WDTO	000044	220#																		
XFLAG	000005R	185#																		

. ABS. 000000 000
 001754 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

XKWEB0,XKWEB0/SOL/CRF:SYN=DDXCOM,XKWEB0
 RUN-TIME: 1 2 3 SECONDS
 RUN-TIME RATIO: 31/3=7.7
 CORE USED: 7K (13 PAGES)