

.REM.

IDENTIFICATION

PRODUCT CODE: AC-F422A-MC
PRODUCT NAME: CXMNCA0 MNCKW MODULE
PRODUCT DATE: FEBRUARY 1979
MAINTAINER: RAY SHOOP

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) 1979, 1979 DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT

THE MNC IS AN IOMOD THAT EXERCISES THE MNCKW 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. UP TO 8 MNCKW WILL BE EXERCISED WITH THIS MODULE. THE "MNA" (A/D) MODULE CAN BE ENABLED TO USE THE CLOCK TO START THE A/D. IF YOU HAVE ENABLED "MNA" TO USE THE CLOCK, YOU SHOULD DESELECT THIS EXERCISER MODULE.

2.0 REQUIREMENTS

HARDWARE: ONE MNCKW (CLOCK).

STORAGE: MNC REQUIRES:
DECIMAL WORDS: 508
OCTAL WORDS: 775
OCTAL BYTES: 1772

3.0 PASS DEFINITION

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

5.0 CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 171020, VECTOR 440, BR1: 4

DEV CNT: 1, SR1: 0

REQUIRED PARAMETERS:

NONE.

6.0 DEVICE/OUTPUT SET-UP

THE FRONT PANEL SWITCHES MUST SELECT THE SCHMITT TRIGGER INPUT
(PULL OUT "S11" AND "S12" SWITCHES AND ROTATE TO THE END).

7.0 MODULE OPERATION

TEST SEQUENCE:

1. (START) BIT EXERCISE CSR, PRESFT REGISTER OF CLOCK.
2. (RSTRT) 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 RATE: 1MHZ.
- B. COUNT TEST CLOCK RATE: 100KHZ.
- C. COUNT TEST CLOCK RATE: 10KHZ.
- D. COUNT TEST CLOCK RATE: 1KHZ.
- E. COUNT TEST CLOCK RATE: 100HZ.
- F. COUNT TEST CLOCK 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

*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.

```

180 000000 IOMOD <MNCA >,171020,440,4,0,0,60,,0
181 000000 MODULE 140000,MNCA ,171020,440,4,0,0,60,,0
182 .TITLE MNCA DEC/X11 SYSTEM EXERCISER MODULE
183 ; DDXCOM VERSION 6 23-MAY-78
184 .LIST BIN
185 ;*****
186 000000 BEGIN;
187 000000 047115 040503 040 MODNAM: .ASCII /MNCA / ;MODULE NAME,
188 000005 000 XFLAG: .BYTE OFEN ;USED TO KEEP TRACK OF WBUFF USAGE
189 000006 171020 ADDR: 171020+0 ;1ST DEVICE ADDR,
190 000010 000440 VECTOR: 440+0 ;1ST DEVICE VECTOR,
191 000012 200 BR1: .BYTE PRTY4+0 ;1ST BR LEVEL,
192 000013 000 BR2: .BYTE PRTY0+0 ;2ND BR LEVEL,
193 000014 000001 DVID1: 0+1 ;DEVICE INDICATOR 1,
194 000016 000000 SR1: OFEN ;SWITCH REGISTER 1
195 000020 000000 SR2: OPEN ;SWITCH REGISTER 2
196 000022 000000 SR3: OPEN ;SWITCH REGISTER 3
197 000024 000000 SR4: OPEN ;SWITCH REGISTER 4
198 ;*****
199 000026 140000 STAT: 140000 ;STATUS WORD,
200 000030 000254 INIT: START ;MODULE START ADDR,
201 000032 000224 SPOINT: MODSP ;MODULE STACK POINTER,
202 000034 000000 PASCNT: 0 ;PASS COUNTER,
203 000036 000074 ICNT: 60 ;# OF ITERATIONS PER PASS=60,
204 000040 000000 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
205 000042 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
206 000044 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
207 000046 000000 SOPPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
208 000050 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
209 000052 000000 SYSCNT: 0 ;# OF SYS ERORS ACCUMULATED
210 000054 000000 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
211 000056 000000 CONFIG: ;RESERVED FOR MONITOR USE
212 000056 000000 RES1: 0 ;RESERVED FOR MONITOR USE
213 000060 000000 RES2: 0 ;RESERVED FOR MONITOR USE
214 000062 000000 SVR0: OPEN ;LOC TO SAVE R0,
215 000064 000000 SVR1: OPEN ;LOC TO SAVE R1,
216 000066 000000 SVR2: OPEN ;LOC TO SAVE R2,
217 000070 000000 SVR3: OPEN ;LOC TO SAVE R3,
218 000072 000000 SVR4: OPEN ;LOC TO SAVE R4,
219 000074 000000 SVR5: OPEN ;LOC TO SAVE R5,
220 000076 000000 SVR6: OPEN ;LOC TO SAVE R6,
221 000100 000000 CSRA: OPEN ;ADDR OF CURRENT CSR,
222 000102 000000 SRADR: ;ADDR OF GOOD DATA, OR
223 000102 000000 ACSR: OPEN ;CONTENTS OF CSR,
224 000104 000000 WBSADR: ;ADDR OF BAD DATA, OR
225 000104 000000 ASTAT: OPEN ;STATUS REG CONTENTS,
226 000106 000000 ERR'YP: ;TYPE OF ERROR
227 000106 000000 ASB: OPEN ;EXPECTED DATA,
228 000110 000000 AWAS: OPEN ;ACTUAL DATA,
229 000112 000254 RSTRT: RESTPT ;RESTART ADDRESS AFTER END OF PASS
230 000114 000000 WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
231 000116 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
232 000120 000000 INTK: OPEN ;# OF INTERRUPTS PER ITERATION
233 000122 000000 IDNUM: 0 ;MODULE IDENTIFICATION NUMBER=0
234 .REPT SPSIZ ;MODULE STACK STARTS HERE,
235 .LIST

```

```

236 .WORD 0
237 .LIST
238 .ENDR
239 000224 MODSP:
240 ;*****
241 000224 000001 TEMP: 1 ;MASK OF CURRENT UNIT
242
243 ;MODULE REQUIRED REGISTERS = SET UP BY THIS MODULE,
244
245 000226 171020 ASP: .WORD 171020 ;CLOCK A STATUS REG,
246 000230 171022 ABR: .WORD 171022 ;CLOCK A BUFFER REG,
247
248 000232 000440 AVECT: .WORD 440 ;CLOCK A INTERRUPT VECTOR,
249 000234 000442 BVECT: .WORD 442 ;CLOCK B INTERRUPT VECTOR,
250
251 000236 000444 BVECT: .WORD 444 ;CLOCK INTERRUPT VECTOR,
252 000240 000446 BVECT2: .WORD 446
253
254 000242 000001 RATEP: .WORD 1 ;POINTS TO CURRENT RATE
255 000244 000000 OFF: .WORD 0 ;OFFSET TO TAKE US TO RATE ROUTINE
256 000246 000000 RAN1: .WORD 0 ;RANDOM NUMBER,
257 000250 000000 RAN2: .WORD 0 ;RANDOM NUMBER,
258 000252 000000 AIFLG: .WORD 0 ;FLAG TO SHOW THAT CLOCK A HAS INTERRUPTED,
259
260 000254 RESTPT:
261 000254 004767 000002 START: JSR PC,START0 ;PRIME THE ADDRESSES
262 000260 000442 BR LOG1 ;RUN LOGIC TEST
263
264 000262 012767 000001 177734 START0: MOV #BIT0,TEMP ;LOAD CURRENT UNIT MASK
265 000270 016767 177512 177730 START1: MOV ADDR,ASP ;GET BASE ADDR,
266 000276 016767 177506 177726 MOV VECTOR,AVECT ;GET BASE VECTOR ADDR,
267 000304 016767 177716 177716 START2: MOV ASR,ABR ;NOW WE'RE GONNA FIX
268 000312 062767 000002 177710 ADD #2,ABR ;ALL CLOCK ADDRESSES BASED ON ASP,
269 000320 016700 177706 MOV AVECT,R0 ;NOW FIX VECTOR ADDRESSES
270 000324 062700 000004 ADD #4,R0
271 000330 010067 177702 MOV R0,BVECT
272 000334 016767 177672 177672 MOV AVECT,AVECT2
273 000342 062767 000002 177664 ADD #2,AVECT2
274 000350 016767 177662 177662 MOV BVECT,BVECT2
275 000356 062767 000002 177654 ADD #2,BVECT2
276 000364 000207 RTS PC ;EXIT

```

```
277 ;*
278 ;*LOGIC TEST #1 BE SURE A CLOCK EXISTS AT THE
279 ;*SPECIFIC ADDR. IF NO CLOCK, THEN A
280 ;*DEC/X11 SYS EPROR WILL OCCUP.
281 ;*
282
283 000366 005777 177634 LOG1: TST 0ASR ;ADDRESS THE CLOCK, IF SYS ERROR
284 ;OCCURS, THEN CLOCK DID NOT
285 ;RETURN SLAVE-SYN WHEN
286 ;ADDRESSED.
287
288 ;*
289 ;*LOGIC TEST #2. MAKE SURE CLOCK CSR BITS
290 ;*14,11,6,5,2 AND 0 CAN BE SET + CLEARED.
291 ;*
292
293 000372 012767 044125 177504 LOG2: MOV #044125,ASTAT ;GENERATE + RECORD PATTERN TO BE USED,
294 000400 016777 177500 177620 MOV ASTAT,0ASR ;SET THEM IN CSR OF CLOCK A.
295 000406 017767 177614 177466 MOV 0ASR,ACSR ;READ THEM BACK
296 000414 026767 177464 177460 CMP ASTAT,ACSR ;DID THEY ALL SET?
297
298 000422 001412 BEQ 20 ;YES - GO TO NEXT TEST.
299 000424 104407 000000 BREAK0,BEGIN ;TEMPORARY RETURN TO MONITOR....
300 000430 104407 000000 BREAK0,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
301 000434 016767 177566 177436 10: MOV ASR,CSRA ;RECORD CSR'S ADDR
302 ;*****
303 000442 104405 000000 000000 HRDR0,BEGIN,NULL ;PATTERN 044125 FAILED
304 ;*****
305 000450 005077 177552 20: CLR 0ASR ;TRY CLEARING THE BITS
306 000454 017767 177546 177420 MOV 0ASR,ACSR ;READ IT BACK.
307 000462 001414 BEQ LOG3 ;IF ZERO CSR GOOD.
308 000464 104407 000000 BREAK0,BEGIN ;TEMPORARY RETURN TO MONITOR....
309 000470 104407 000000 BREAK0,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
310 000474 005067 177404 30: CLR ASTAT ;EXPECT ZERO CSR.
311 000500 016767 177522 177372 MOV ASR,CSRA ;RECORD CSR'S ADDR.
312 ;*****
313 000506 104405 000000 000000 HRDR0,BEGIN,NULL ;CSR FAILED TO CLEAR
314 ;*****
```

```
315 ;*
316 ;*LOGIC TEST #3. MAKE SURE CLOCK CSR BITS
317 ;*13,5,3 AND 1 CAN BE SET + CLEARED
318 ;*
319
320 000514 012767 020052 177362 LOG3: MOV #020052,ASTAT ;GENERATE + RECORD PATTERN TO BE USED,
321 000522 016777 177356 177476 MOV ASTAT,0ASR ;SET THEM IN CSR OF CLOCK A.
322 000530 017767 177472 177344 MOV 0ASR,ACSR ;READ THEM BACK
323 000536 026767 177342 177336 CMP ASTAT,ACSR ;DID THEY ALL SET?
324 000544 001412 BEQ 20 ;YES - GO TO NEXT TEST.
325 000546 104407 000000 BREAK0,BEGIN ;TEMPORARY RETURN TO MONITOR....
326 000552 104407 000000 BREAK0,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
327 000556 016767 177444 177314 10: MOV ASR,CSRA ;RECORD CSR'S ADDR.
328 ;*****
329 000564 104405 000000 000000 HRDR0,BEGIN,NULL ;CSR PATTERN 020052 FAILED
330 ;*****
331 000572 005077 177430 20: CLR 0ASR ;TRY CLEARING THE BITS
332 000576 017767 177424 177276 MOV 0ASR,ACSR ;READ IT BACK.
333 000604 001414 BEQ LOG4 ;IF ZERO CSR GOOD.
334 000606 104407 000000 BREAK0,BEGIN ;TEMPORARY RETURN TO MONITOR....
335 000612 104407 000000 BREAK0,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
336 000616 005067 177262 30: CLR ASTAT ;EXPECT ZERO CSR.
337 000622 016767 177400 177250 MOV ASR,CSRA ;RECORD CSR'S ADDR.
338 ;*****
339 000630 104405 000000 000000 HRDR0,BEGIN,NULL ;CSR FAILED TO CLEAR
340 ;*****
341 ;*
342 ;*LOGIC TEST #4. MAKE SURE CLOCK BUFFER REG
343 ;*PATTERN 125252 CAN BE SET + CLEARED.
344 ;*
345
346 000636 012767 125252 177240 LOG4: MOV #125252,ASTAT ;GENERATE + RECORD PATTERN TO BE USED,
347 000644 016777 177234 177356 MOV ASTAT,0ASR ;SET THEM IN BUFFER REG OF CLOCK.
348 000652 017767 177352 177222 MOV 0ASR,ACSR ;READ THEM BACK
349 000660 026767 177220 177214 CMP ASTAT,ACSR ;DID THEY ALL SET?
350 000666 001412 BEQ 20 ;YES - GO TO NEXT TEST.
351 000670 104407 000000 BREAK0,BEGIN ;TEMPORARY RETURN TO MONITOR....
352 000674 104407 000000 BREAK0,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
353 000700 016767 177324 177172 10: MOV ASR,CSRA ;RECORD CSR'S ADDR.
354 ;*****
355 000706 104405 000000 000000 HRDR0,BEGIN,NULL ;BUFFER REG PATTERN 125252 FAILED
356 ;*****
357 000714 005077 177310 20: CLR 0ASR ;TRY CLEARING THE BITS
358 000720 017767 177304 177154 MOV 0ASR,ACSR ;READ IT BACK.
359 000726 001414 BEQ LOG5 ;IF ZERO BUFFER GOOD.
360 000730 104407 000000 BREAK0,BEGIN ;TEMPORARY RETURN TO MONITOR....
361 000734 104407 000000 BREAK0,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
362 000740 005067 177140 30: CLR ASTAT ;EXPECT ZERO BUFFER.
363 000744 016767 177260 177126 MOV ASR,CSRA ;RECORD ADDR. OF BUFFER REG.
364 ;*****
365 000752 104405 000000 000000 HRDR0,BEGIN,NULL ;BUFFER REG FAILED TO CLEAR
366 ;*****
367 ;*LOGIC TEST #5. MAKE SURE CLOCK BUFFER REG
368 ;*PATTERN 052525 CAN BE SET + CLEARED
369 000760 012767 052525 177116 LOG5: MOV #052525,ASTAT ;GENERATE + RECORD PATTERN TO BE USED,
370 000766 016777 177112 177234 MOV ASTAT,0ASR ;SET THEM IN BUFFER OF CLOCK A.
```

```

371 000774 017767 177230 177100 MOV @ABR,ACSR ;READ THEM BACK
372 001002 026767 177076 177072 CMP ASTAT,ACSR ;DID THEY ALL SET?
373 001010 001412 EQ 26 ;YES - GO TO NEXT TEST.
374 001012 104407 000000 BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
375 001016 104407 000000 BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
376 001022 016767 177202 177050 181 MOV @RR,CSRA ;RECORD BUFFER REG ADDR.
377
378 001030 104405 000000 000000 ;*****
379 ;*****
380 001036 005077 177166 201 CLR @ABP ;TRY CLEARING THE BITS
381 001042 017767 177162 177032 MOV @ARF,ACSF ;READ IT BACK.
382 001050 001414 BEQ @RST0 ;IF ZERO BUFFER GOOD,
383 001052 104407 000000 BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR,...
384 001056 104407 000000 BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
385 001062 005067 177016 381 CLR ASTAT ;EXPECT ZERO BUFFER,
386 001066 016767 177136 177004 MOV @AR,CSRA ;RECORD BUFFER REG A ADDR.
387 ;*****
388 001074 104405 000000 000000 ;*****
389 ;*****
390 001102 006367 177116 REST0: ASL TEMP ;CHECK FOR NEXT UNIT
391 001106 022767 000000 177110 CMP @R10,TEMP ;TEST IF VALID UNIT
392 001114 001420 BEQ @RST1 ;RR IF NOT
393 001116 036767 177102 176670 BIT TEMP,DVID1 ;TEST IF THIS UNIT IS SELECTED
394 001124 001414 BEQ @RST1 ;RR IF NOT
395 001126 005077 177074 CLR @ASR ;ENSURE CLOCK IS STOPPED
396 001132 022767 000000 177066 ADD #4,ASR ;UPDATE STATUS ADDRESS
397 001140 062767 000010 177064 ADD #10,AVECT ;UPDATE VECTOR ADDRESS
398 001146 004767 177132 JSR PC,START2 ;FIX OTHER VALUES
399 001152 000167 177210 JMP LOG1 ;CONTINUE AT LOG1
400 001156 012767 000100 177056 REST1: MOV #R10,RATEP ;FIRST PASS THRU LOOP, OFFSET=0, RATES WILL=1.
401 001164 012777 001574 177040 PEST2: MOV #INSRV,RAVECT ;SET UP CLOCK A'S INTER. VECTOR.
402 001172 116777 176614 177034 REST3: MOV @R1,RAVECT2 ;SET PRIORITY ON CLOCK A'S INTR.
403 001200 106167 177036 LOOP: POLR RATEP ;GET NEXT RATE.
404 001204 100005 BPL 18 ;IF NOT END THEN CONTINUE,
405 001206 005067 177032 CLR OFF ;CLEAR THE OFFSET
406 001212 012767 000001 177022 MOV #1,RATEP ;LOOK AT FIRST RATE
407 001220 062767 000002 177016 181 ADD #2,OFF
408 001226 022767 000016 177010 CMP #16,OFF ;TEST IF #16
409 001234 001761 BEQ LOOP ;RR
410 001236 036767 177000 176552 BIT RATEP,SR1 ;IS THIS RATE INHIBITED?
411 001244 001355 BNE LOOP
412 001246 005067 177000 CLR @FLG ;CLR FLAG INDICATING CLOCK A HAS INTERRUPTED.
413 001252 016701 176766 MOV OFF,R1 ;PICK UP OFFSET
414 001256 000171 001262 JMP @LISTP(R1) ;GO SET THE RATE + START THE CLOCK.

```

```

415 ;THE FOLLOWING (LISTP) ARE POINTERS TO VARIOUS RATE
416 ;ROUTINES. THEY ARE USED BY "LOOP". "LOOP" GENERATES
417 ;AN OFFSET OF A RATE WE WISH TO EXERCISE. THE OFFSET
418 ;IS STORED IN R1. WE INDEX "LISTP" BY R1 (JMP @LISTP(R1))
419 ;TO GET THE ADDRESS OF THE RATE ROUTINE TO EXERCISE.
420 ;
421
422 LISTP: .WORD 1 ;
423 .WORD RATE0 ;POINTER TO 1MHZ ROUTINE
424 .WORD RATE1 ;POINTER TO 100KHZ ROUTINE
425 .WORD RATE2 ;POINTER TO 10KHZ ROUTINE
426 .WORD RATE3 ;POINTER TO 1KHZ ROUTINE
427 .WORD RATE4 ;POINTER TO 100HZ ROUTINE
428 .WORD RATE5 ;POINTER TO RANDOM ROUTINE
429 .WORD 1 ;
430
431 ;
432 ;THE FOLLOWING (RATEAL) ARE THE PRESET VALUES THAT THE
433 ;VARIOUS RATE ROUTINES NEED. THEY ARE LOADED INTO
434 ;CLOCK A'S PRESET BUFFER. "RATEAL" IS INDEXED BY
435 ;AN OFFSET IN R1 BY THE RATE ROUTINES TO GET THE
436 ;PRESET VALUE
437 ;
438
439 RATEAL: .WORD 1 ;OFFSET ZERO, NO RATE.
440 .WORD -50000 ;VALUE FOR 1MHZ PRESET.
441 .WORD -50000 ;PRESET VALUE FOR 100 KHZ
442 .WORD -10000 ;PRESET VALUE FOR 10 KHZ
443 .WORD -1000 ;PRESET VALUE FOR 1 KHZ
444 .WORD -100 ;PRESET VALUE FOR 100 HZ
445 .WORD 0 ;PRESET VALUE FOR RANDOM
446 .WORD 1 ;
447
448 ;
449 ;THE FOLLOWING (RSAL) IS USED BY THE RANDOM
450 ;RATE ROUTINE (RATE5). THEY ARE THE VALUES NEEDED
451 ;TO BE PUT INTO THE CLOCK'S CSR FOR A PARTICULAR RATE.
452 ;
453
454 RSAL: .WORD 0 ;OFFSET ZERO, NO RATE.
455 .WORD 113 ;11 MHZ, GO., MODE 1
456 .WORD 123 ;100 KHZ, GO., MODE 1
457 .WORD 131 ;10 KHZ, GO.
458 .WORD 141 ;1 KHZ, GO.
459 .WORD 151 ;100 HZ, GO.
460 .WORD 0 ;

```

```

461 ;*THIS ROUTINE PRESETS CLOCK A FOR
462 ;*1 MHZ RATE CLOCK A INTR IN 1/20 SEC., 25 TIMES.
463 ;*
464 ;*
465
466 001340*          RATE0:
467 001340* 005077 176662 CLR  #ASR          ;CLEAR CLOCK A.
468 001344* 016177 001302* 176656 MOV  RATEAL(R1),#ABR ;PRESET COUNT IN CLOCK A.
469 001352* 016177 001322* 176646 MOV  RSAL(R1),#ASR  ;START CLOCK A.
470 ;NOW WAIT FOR INTERRUPT
471 001360* 104400 EXIT# ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
472
473 ;*
474 ;*THIS ROUTINE PRESETS CLOCK A FOR
475 ;*100 KHZ RATE CLOCK A INTR IN .5 SEC., TWICE.
476 ;*
477 ;*
478
479 001362*          RATE1:
480 001362* 005077 176640 CLR  #ASR          ;CLEAR CLOCK A.
481 001366* 016177 001302* 176634 MOV  RATEAL(R1),#ABR ;PRESET COUNT IN CLOCK A.
482 001374* 016177 001322* 176624 MOV  RSAL(R1),#ASR  ;START CLOCK A.
483 ;NOW WAIT FOR INTERRUPT.
484 001402* 104400 EXIT# ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
485
486 ;*
487 ;*THIS ROUTINE PRESETS CLOCK A FOR
488 ;*10 KHZ RATE CLOCK A INTR IN 1.0 SEC.
489 ;*
490 ;*
491
492 001404*          RATE2:
493 001404* 005077 176616 CLR  #ASR          ;CLEAR CLOCK A.
494 001410* 016177 001302* 176612 MOV  RATEAL(R1),#ABR ;PRESET COUNT IN CLOCK A.
495 001416* 016177 001322* 176602 MOV  RSAL(R1),#ASR  ;START CLOCK A.
496 ;NOW WAIT FOR INTERRUPT.
497 001424* 104400 EXIT# ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
498
499 ;*
500 ;*THIS ROUTINE PRESETS CLOCK FOR
501 ;*1 KHZ RATE CLOCK A INTR IN 1.0 SEC.
502 ;*
503 ;*
504
505 001426*          RATE3:
506 001426* 005077 176574 CLR  #ASR          ;CLEAR CLOCK A.
507 001432* 016177 001302* 176570 MOV  RATEAL(R1),#ABR ;PRESET COUNT IN CLOCK A.
508 001440* 016177 001322* 176560 MOV  RSAL(R1),#ASR  ;START CLOCK A.
509 ;NOW WAIT FOR INTERRUPT.
510 001446* 104400 EXIT# ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

```

```

511 ;*THIS ROUTINE PRESETS CLOCK FOR
512 ;*100 HZ RATE CLOCK A INTR IN 1.0 SEC.
513 ;*
514 ;*
515
516 001450*          RATE4:
517 001450* 005077 176552 CLR  #ASR          ;CLEAR CLOCK A.
518 001454* 016177 001302* 176546 MOV  RATEAL(R1),#ABR ;PRESET COUNT IN CLOCK A.
519 001462* 016177 001322* 176536 MOV  RSAL(R1),#ASR  ;START CLOCK A.
520 ;NOW WAIT FOR INTERRUPT.
521 001470* 104400 EXIT# ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
522
523 ;*THIS ROUTINE PRESETS CLOCK FOR
524 ;*RANDOM RATES
525
526 001472* 004767 000246 RATE5: JSR  PC,RANDOM  ;GET 2 RANDOM NUMBERS.
527
528 001476* 042767 177771 176542 BIC  #177771,RANA  ;MAKE NUMBER < 10.
529 001504* 042767 177771 176536 BIC  #177771,RANB  ;MAKE 2ND NUMBER < 10.
530
531 ;NUMBERS MUST BE 2, 4, OR 6
532
533 001512* 005767 176530 30: TST  PANA          ;IS NUMBER ZERO?
534 001516* 001003 RNE  4#           ;NO - GO AHEAD.
535 001520* 062767 000002 176520 JDD  #2,PANA      ;MAKE IT NON-ZERO.
536 001526* 005767 176516 40: TST  RANB          ;IS NUMBER ZERO?
537 001532* 001003 BNE  5#           ;NO GO AHEAD.
538 001534* 062767 000002 176506 JDD  #2,RANB      ;MAKE IT NON-ZERO.
539 001542*
540 001542* 005077 176460 CLR  #ASR          ;CLEAR CLOCK A
541 001546* 016701 176474 MOV  RANA,R1      ;RECORD THE OFFSET
542 001552* 010167 176466 MOV  R1,OFF
543 001556* 016177 001302* 176444 MOV  RATEAL(P1),#ABR ;PRESET CLOCK A.
544 001564* 016177 001322* 176434 MOV  RSAL(R1),#ASR  ;START CLOCK A.
545
546 001572* 104400 EXIT# ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT

```


ICON	000036R	203#												
ICOUNT	000040R	204#												
IDNUM	000122R	233#												
INIT	000030R	200#												
INSEPV	001574R	401	540#											
INTR	000120R	232#												
LISTP	001262P	414	422#											
LOG1	000366P	262	283#	399										
LOG2	000372R	293#												
LOG3	000514R	307	320#											
LOG4	000636R	333	346#											
LOG5	000760R	359	369#											
LOOP	001200R	403#	409	411										
MAP22#	104416	241#												
MODNAM	000000R	187#												
MODSP	000224R	201	239#											
MSGN#	104403	241#												
MSG0#	104402	241#												
MSG#	104401	241#												
NULL	000000	241#	303	313	329	339	355	365	378					
OFF	000244R	255#	405#	408	413	542#	550	555						
OPEN	000000	188	194	195	196	197	214	215	216	217	218	219	220	221
		223	225	227	228	230	231	232	241#					
OTOA#	104420	241#												
PASCNT	000034R	202#												
PIRO#	000004	241#	562											
POPSP	005726	241#												
POPSP2	022626	241#												
PRTY	000000	241#												
PRTY0	000000	192	241#											
PRTY1	000040	241#												
PRTY2	000100	241#												
PRTY3	000140	241#												
PRTY4	000200	191	241#											
PRTY5	000240	241#												
PRTY6	000300	241#												
PRTY7	000340	241#												
PS	177776	241#												
PSW	177776	241#												
PUSH	005746	241#												
PUSH2	024646	241#												
RANA	000246R	256#	528#	533	535#	541	580	581#	582*					
RANB	000250R	257#	529#	536	538#	580*	582	583#						
RANDOM	001744R	526	580#											
RAND#	104417	241#												
RANNUM	000054R	210#												
RATEAL	001302R	439#	468	481	494	507	518	543						
RATEP	000242P	254#	400#	403#	406#	410								
RATE0	001340R	423	466#											
RATE1	001362P	424	479#											
RATE2	001404R	425	492#											
RATE3	001426R	426	505#											
RATE4	001450R	427	516#											
RATE5	001472R	428	526#											
RESTR1	000254R	229	260#											
RESTR0	001102R	382	390#											

REST1	001156R	392	394	400#										
REST2	001164P	401#	570											
REST3	001172R	402#	576											
RES1	000056R	212#												
RES2	000060P	213#												
RSAL	001322R	454#	469	482	495	508	519	544						
RSTRT	000112P	229#												
SBADR	000102R	222#												
SOFcnt	000042P	205#												
SOFER#	104406	241#												
SOPFAS	000046R	207#												
SPOINT	000032R	201#												
SPSIZ	000040	1	234											
SR1	000016R	194#	410											
SR2	000020R	195#												
SR3	000022R	196#												
SR4	000024R	197#												
START	000254R	200	261#											
START0	000262R	261	264#											
START1	000270R	265#												
START2	000304P	267#	398	575										
STAT	000026R	199#												
SVR0	000062P	214#												
SVR1	000064R	215#												
SVR2	000066R	216#												
SVR3	000070P	217#												
SVR4	000072P	218#												
SVR5	000074R	219#												
SVR6	000076R	220#												
SYS cnt	000052P	209#												
TEMP	000224R	241#	264#	390#	391	393	564#	565	571					
TRPDFD	000022	241#												
VECTOR	000010P	190#	266											
WASADR	000104R	224#												
WDFR	000116R	231#												
WDTO	000114P	230#												
XFLAG	000005R	188#												

. ABB. 000000 000
 001772 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

PACK:XMNCA0,PACK:XMNCA0/BOL/CRF;SYN=NDXCOM,PACK:XMNCA0
 RUN-TIME: 3 5 .7 SECONDS
 RUN-TIME RATIO: 43/10=4.1
 CORE USED: 7K (13 PAGES)