

RX211,RXV11

UTILITY DRIVER
CZRXC A0

AH-E510A-MC

COPYRIGHT 1978

FICHE 1 OF 1

JAN 1979

digital

MADE IN USA

.REM &

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

IDENTIFICATION

PRODUCT CODE: AC-E509A-MC
PRODUCT NAME: CZRXCA0 RX02 UTIL DRVR
DATE CREATED: DEC 1978
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: L.S. PRUCHA

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 DIGITAL'S 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) 1978, BY DIGITAL EQUIPMENT CORPORATION

49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	HARDWARE
2.2	STORAGE
3.0	LOADING PROCEDURE
4.0	STARTING PROCEDURE
5.0	CONSOLE SWITCHES
6.0	OPERATION
6.1	FUNCTION CODES
6.2	DATA FORMATS
6.3	DENSITY
6.4	UNIT SELECT
6.5	SIDE SELECT
6.6	HARDWARE REGISTERS
6.7	DEVICE ERROR CODES
7.0	PROGRAM DESCRIPTION
7.1	FLOW
7.2	TABLES
8.0	LISTING INDEX
8.1	LISTING

94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134

1.0 ABSTRACT

THIS PROGRAM IS INTENDED AS A BRUTE FORCE ROUTINE TO EXECUTE AN OPERATION OR SERIES OF OPERATIONS, CONTINUOUSLY REGARDLESS OF THE RESULTS OF THE OPERATION. BECAUSE OF THE COMPLEXITY OF THE RX01,02 FLOPPY DISK SYSTEM AS OPERATED ON THE UNIBUS, IT IS NOT ALWAYS POSSIBLE TO PROVIDE FOR EVERY CONTINGENCY IN THE NORMAL PROGRAMS THEREFORE THIS UTILITY DRIVER WILL ALLOW AN OPERATOR TO EXECUTE ANYTHING DESIRED IN ANY ORDER. THERE ARE NO ERROR CHECKS OR PRINTOUTS MADE, AND ANY VARIATION FROM PRESET SEQUENCES AND VALUES ARE MADE BY CHANGING THE APPROPRIATE MEMORY LOCATIONS.

2.0 REQUIREMENTS

2.1 HARDWARE

- 1. ANY PDP-11 PROCESSOR
2. RX11, RXV11, RX211, OR RXV21 BUS INTERFACE
3. RX01 OR RX02 DEVICE

2.2 STORAGE

THIS PROGRAM REQUIRES AT LEAST 2K OF CORE

3.0 LOADING PROCEDURE

USE STANDARD BINARY LOADING PROCEDURE

4.0 STARTING PROCEDURE

THE PROGRAM IS ALWAYS STARTED AT LOCATION 200(8)

136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171

5.0 CONSOLE SWITCH SETTINGS <SEE NOTE BELOW>

HARDWARE SWITCH REG OR IF SOFTWARE SWITCH REG --> (SFSWR LOC:760)

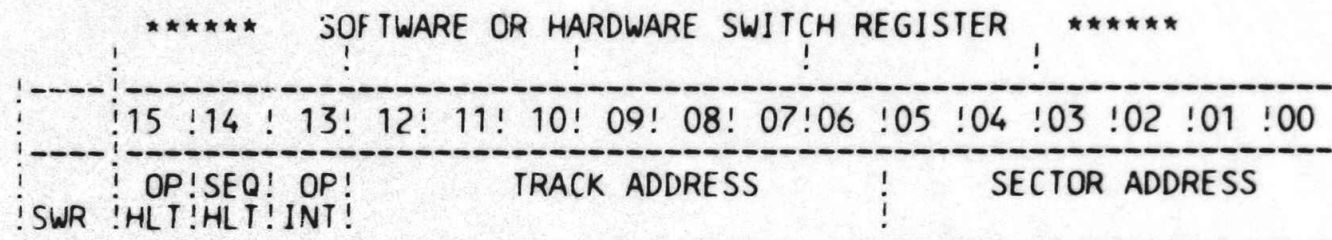
SW 15=1 (100000) - STOP ON EACH OPERATION
=0 - CONTINUE

SW 14=1 (040000) - STOP ON END OF SEQUENCE
=0 - CONTINUE

SW 13=1 (020000) - INITIALIZE DEVICE BEFORE EACH OPERATION
=0 - NO INTALIZATION BETWEEN OPERATIONS

SW 12=6 (6-0,RXTA) - TRACK ADDRESS

SW 5=0 (4-0,RXSA) - SECTOR ADDRESS



NOTE: IF PROCESSOR DOES NOT HAVE HARDWARE SWITCH REGISTER AT LOCATION 177570 THEN THE PROGRAM WILL USE ITS OWN INTERNAL SOFTWARE SWITCH REGISTER AT 760. THE PROGRAM WILL DETECT THE NON-EXISTENCE OF A HARDWARE SWITCH REGISTER.

173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226

6.0 OPERATION

THE PROGRAM OPERATION IS QUITE SIMPLE, BUT DOES REQUIRE THE OPERATOR TO HAVE KNOWLEDGE OF THE RX01,02 FLOPPY DISK SYSTEM AS OPERATED BY RX11,RX211 UNIBUS INTERFACE. THE OPERATOR MUST BE ABLE TO DECIDE WHICH SEQUENCE OF OPERATIONS IS REQUIRED, AND WHAT VALUES TO ASSIGN TO THE VARIOUS PARAMETERS REQUIRED TO EXECUTE THEM. THE OPERATION SEQUENCE IS SET UP BY LOADING A TABLE WITH THE FUNCTION CODES OF THE DESIRED OPERATIONS AND SETTING THE NUMBER OF OPERATIONS IN A COUNTER. THE PROGRAM IS SET UP TO OPERATE IN SINGLE DENSITY MODE, WITH UNIT 0 PRESELECTED, AND TO DO AN EMPTY BUFFER OPERATION. THE OPERATION SEQUENCE WILL BE EXECUTED CONTINUOUSLY IF LOADED AND STARTED AT 200(8) WITH NO CHANGES MADE AND SWITCHES 14 AND 15 SET AT ZERO(0). THE FOLLOWING IS THE LIST OF PARAMETERS WHICH MAY BE VARIED AND A DESCRIPTION OF EACH ALONG WITH THEIR CORE LOCATION:

PARAMETER	LOCATION	DESCRIPTION
-----	-----	-----
RXCS	600	ADDRESS OF RX CONTROL + STATUS REG
RXDB	602	ADDRESS OF RX DATA BUFFER REG
PSW	604	ADDRESS OF PROCESSOR STATUS WORDS
SWR	606	ADDRESS OF SWITCH REGISTER WORD
SETCS	700	SET PART OF RXCS WORD - BIT#4 UNIT SELECT BIT#6 INTERRUPT ENABLE BIT#8 DENSITY SELECT (RX02,XX ONLY) BIT#9 SIDE SELECT (RXXX ONLY)
WRDCNT	702	WORD COUNT- SET NUMBER OF WORDS TO BE TRANSFERRED RX02,XX (RX01-N/A)
RDYDLY	704	READY DELAY- THIS DELAY VALUE IS USED BY THE PROGRAM TO ESTABLISH A MAXIMUM TIME TO AWAIT THE COMPLETION OF AN OPERATION BEFORE PROCEEDING TO THE NEXT. ** (DEFAULT IS APPROX 435 MS FOR PDP-11/20) **
RDYDX	706	READY MULTIPLIER- IF THE VALUE SET INTO 704 DOES NOT ALLOW ENOUGH TIME, INCREASE THE SIZE OF THE MULTIPLIER. EACH INCREMENT OF THE MULTIPLIER WILL CAUSE THE 704 DELAY TO BE EXECUTED THAT MANY MORE TIMES.
OPNUM	710	OPERATION NUMBER- THIS IS THE NUMBER OF OPERATIONS TO BE PERFORMED IN A SEQUENCE AND SHOULD REFLECT THE NUMBER OF OPERATIONS SET INTO THE FUNCTION TABLE. (16 MAXIMUM)

228			
229			
230	FUNTAB	712-750	FUNCTION TABLE-
231			THIS TABLE (CONSISTING OF 16 WORD LOCATIONS) IS TO
232			BE LOADED WITH THE FUNCTION CODES FOR EACH
233			OPERATION TO BE PERFORMED IN SEQUENCE. THE
234			NUMBER OF ENTERIES MAY BE FROM ONE(1) TO
235			SIXTEEN(16). MAKE SURE THAT THE NUMBER OF
236			FUNCTION CODES SET IN THE TABLE IS REFLECTED
237			BY THE NUMBER IN LOCATION 710 (OPNUM).
238			
239	SOFTSR	760	SOFTWARE SWITCH REGISTER - USED FOR
240			SWITCHLESS PROCESSORS.
241			
242	ERRTAB	3700	ERROR TABLE- ERROR REG FOR RESP. FUCTIONS
243	INTAB	4000	INPUT TABLE- RX DATA BUFFER INTO TABLE
244	OUTAB	5000	OUTPUT TABLE- TABLE INTO RX DATA BUFFER
245			
246	XETBA	4400	EXTENDED ERROR TABLE - THIS TABLE CONTAINS
247			RESULTS OF READ ERROR CODE OPERATION IN
248			FUNCTION TABLE - SEE TABLE DESCRIPTIONS BELOW.
249			
250			

NOTE: RXXX + XX ARE REFERENCES TO DEVICE FUTURE EXPANSION.

252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306

6.1 FUNCTION CODES

- 0 = FILL BUFFER
- 1 = EMPTY BUFFER
- 2 = WRITE SECTOR
- 3 = READ SECTOR
- 4 = SET DENSITY(RX02 ONLY)
- 5 = READ STATUS
- 6 = WRITE SECTOR WITH DELETED DATA
- 7 = READ ERROR CODE

6.2 DATA FORMATS (RELATIVE TO DENSITY)

SINGLE DENSITY: 64 WORD/SECTOR RX01,02,XX
 DOUBLE DENSITY: 128 WORDS/SECTOR RX02,XX ONLY

6.3 DENSITY (BIT 8 OF CONTROL STATUS WORD, RX02,XX ONLY)

- 0 = DOUBLE FREQUENCY (OR FM) SINGLE DENSITY
- 1 = MILLER CODE (OR MCM) DOUBLE DENSITY

6.4 UNIT SELECT (BIT 4 OF CONTROL STATUS WORD)

SET TO DEVICE UNIT ADDRESS (0-1)

6.5 SIDE SELECT (BIT9 OF CONTROL STATUS WORD, RXXX ONLY)

SET TO SIDE ADDRESS (0-1)

6.6 HARDWARE REGISTERS

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
RXCS:	ERR	INT	XM	XM	RX2	SID	DEN	TR	IE	DON	DRV	FUN	FUN	FUN	GO		
RXWC:											WORD	COUNT					
RXBA:	BUS				ADDRESS				REGISTER								
RXES:					NXM	WC	SID	DRV	DRV	DEL	DSK	DEN	AC	INT	SID	CRC	
					OVF	#1	#1	RDY	DAT	DEN	ERR	LOW	DON	RDY			
RXDB:	DATA				BUFFER												
RXTA:											0	TRACK ADDRESS					
RXSA:											0	0	0	SECTOR ADDRESS			

NOTE: RXXX + XX ARE REFERENCES TO DEVICE FUTURE EXPANSION.

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
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363

6.7 DEVICE ERROR CODES

KNXDVO=10 /DRIVE 0 FAILED TO SEE HOME ON INITIALIZE. NO ERROR BIT
KNXDV1=20 /DRIVE 1 FAILED TO SEE HOME ON INITIALIZE. NO ERROR BIT
KERTRK=40 /TRIED TO ACCESS A TRACK GREATER THAN 76.
KHOMERR=50 /HOME WAS FOUND BEFORE DESIRED TRACK WAS REACHED.
KSELFER=60 /SELF DIAGNOSTIC ERROR.
KNXHDR=70 /DESIRED SECTOR COULD NOT BE FOUND AFTER LOOKING AT 52 HEADERS.
KWPROT=100 /WRITE FUNCTION ATTEMPTED ON A WRITE PROTECTED DISK.
KTIMERR=110 /MORE THAN 40 MICROSECONDS AND NO SEPCLOCK SEEN.
KNXPRAM=120 /A PREAMBLE COULD NOT BE FOUND.
KNXIDAM=130 /PREAMBLE FOUND BUT NO ID MARK FOUND WITHIN ALLOWABLE TIME.
KNCHCER=140 /CRC ERROR ON WHAT APPEARED TO BE A HEADER. ERROR IS NOT ASSERT
KTKSKER=150 /THE TRACK ADDRESS OF A GOOD HEADER DOES NOT COMPARE WITH THE DE
KXSTRYS=160 /TOO MANY TRIES FOR AN IDAM.
KNODAM=170 /DATA AM NOT FOUND IN ALLOTTED TIME.
KDCRCER=200 /CRC ERROR ON READING THE SECTOR FROM THE DISK.
KMANER=220 /R/W ELECTRONICS FAILED MAINTENANCE MODE TEST.
KWCNOV=230 /WORD COUNT OVERFLOW.
KSTDER=240 /WRONG KEY WORD FOR SET MEDIA DENSITY COMMAND.

7.0 PROGRAM DESCRIPTION

IN ORDER TO MAINTAIN THE CONTINUOUS EXECUTION OF THE
OPERATIONS DESCRIBED THE PROGRAM IS ORGANIZED AS FOLLOWS:

START
DETERMINE IF SWITCHLESS PROCESSOR
DETERMINE IF LSI PROCESSOR
INITIALIZE THE RX
GET RX STATUS WORD (WORD COUNT, UNIT SELECT)
GET SWITCH REGISTER
INITIALIZE RX IF SWITCH 13=1
EXECUTE OPERATION (SET FUNCTION AND FROM OP TABLE AND SET GO=1)
AWAIT END OF OPERATION (READY DELAY)
STOP IF SWITCH 15=1
STOP IF LAST OPERATION IN SEQUENCE AND SWITCH 14=1
POINT TO NEXT FUNCTION CODE IN OP TABLE
JUMP BACK TO GET SWITCH REGISTER.

7.1 FLOW

;BEGINROUTINE [MOD 0.0 CONTROL]
: INITIALIZE STACK
: IF HARDWARE REG
: : THEN
: : SET SWITCH_REG ADDRESS=HARDWARE SWITCH REG ADDRESS
: : ELSE
: : SET SWITCH_REG ADDRESS=SOFTWARE SWITCH REG ADDRESS
: ENDIF
: IF NOT LSI_PROCESSOR
: : THEN

364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419

```

: SET NORMAL PROCESSOR PRI
: ELSE
: SET LSI PROCESSOR PRI + SET LSI_FLAG
ENDIF
CALL OUTPUT [MOD 3.0]-OUTPUT INITIALIZE TO RX (INTWD,CSAD)
CALL INPUT [MOD 1.0]-GET RX CSR (DNWD,CSAD,TYIN,CSWD)
BEGINDO
: CALL INPUT [MOD 1.0]-GET SWITCH REG (SWAD,TYIN,SWWD)
: CALL PROCESS [MOD 7.0]-FORMAT TABLE ENTRY (TE,SETFW,FNWD)
: IF SWR BITB=1
: THEN
: CALL OUTPUT [MOD 3.0]-INITIALIZE RX (CSAD,INITWD)
: ENDIF
: CALL OUTPUT [MOD 3.0]-SEND FUNCTION WORD TO RX (CSAD,FW,CSWD,WC,SWWD,ETAD,TE)
: IF FW [FUNCTION WORD]=EMPTY BUFFER
: THEN
: IF DEVICE=RX02 <RXCS BIT11=0>
: THEN
: SET TYIN=64
: CALL INPUT [MOD 1.0]-GET RX01 INPUT BUFFER (DBAD,TYIN)
: ENDIF
: ENDIF
: CALCULATE ERROR TABLE ADDRESS <ETAD=ETBA+2TE>
: IF ERROR_STATUS_FLAG=1 <ERSTAT=1>
: THEN
: CALL INPUT [MOD 1.0]-GET ERROR WORD (DBAD,TYIN,ERWD)
: SAVE ERROR WORD (ERWD) IN ERROR TABLE AT (ERAD)
: ELSE
: CALL INPUT [MOD 1.0]-GET STATUS WORD (CSAD,CSWD,TYIN)
: IF RX CSR BIT15=1 [ERROR BIT]
: THEN
: CALL INPUT [MOD 1.0]-GET ERROR WORD (DBAD,TYIN,ERWD)
: SAVE ERROR WORD (ERWD) IN ERROR TABLE AT (ERAD)
: ENDIF
: ENDIF
: CLEAR ERROR STATUS FLAG <ERSTAT>
: ADVANCE TABLE ENTRY <TE=TE+1>
: IF SWR BIT15=1
: THEN
: HALT
: IF TABLE ENTRY=TABLE LENGTH <TE=TL>
: THEN
: SET TABLE ENTRY=0
: ENDIF
: ELSE
: IF TABLE ENTRY=TABLE LENGTH <TE=TL>
: THEN
: IF SWR BIT14=1
: THEN
: HALT
: ENDIF
: SET TABLE ENTRY=0 <TE=0>
: ENDIF
: ENDIF
: ENDDO
: ENDRoutine

```

421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471

7.2 TABLES

1. THE DATA INPUT (FROM RX) TABLE IS AT LOCATION 4000.
2. THE DATA OUTPUT (TO RX) TABLE IS AT LOCATION 5000.
3. THE ERROR TABLE IS AT LOCATION 3700.
 - ERRORS ARE STORED IN THIS TABLE, ONE WORD FOR ONE WORD IN SEQUENCE CORRESPONDING WITH FUNCTION CODE TABLE. (SEE FUNCTION TABLE BELOW)
 - OPERATION-ERRORS ARE STORED FOR EACH MAINT READ STATUS FUNCTION SELECTED OR ON ANY OTHER FUNCTION IF AN ERROR OCCURRED. THE LOCATION IS ZEROED IF THESE TWO CONDITIONS ARE NOT MET.
4. THE EXTENDED ERROR CODE TABLE IS AT LOCATION 4400.
 - THE RESULTS OF A READ ERROR CODE ARE STORED IN THIS TABLE.
 - OPERATION - IF A READ ERROR CODE FUNCTION IS USED THEN THE RESULTS ARE STORED IN THIS TABLE WITH A STARTING ADDRESS AS SHOWN IN THE FUNCTION TABLE BELOW. THE NEXT 8 BYTES CONTAIN RESULTS OF THE FUNCTION.
5. THE FUNCTION TABLE IS AT LOCATION 712.

8.0 LISTING INDEX

580	MODULE 0.0 - CONTROL
727	MODULE 1.0 - INPUT
764	MODULE 2.0 - PROCESS
787	MODULE 3.0 - OUTPUT
895	MODULE 3.1 - OUTPUT SINGLE WORD
907	MODULE 3.2 - OUTPUT MULTIPLE WORDS
933	MODULE 1/3.1 - DELAY
959	MOD U.2.1 - WATCH DOG TIMER
984	SET PROCESSOR PRI
997	RX INTERRUPT HANDLER
1006	DATA TABLES

8.1 LISTING

&

```

473 .NLIST TTM
474 .TITLE RX01,2,X UTILITY DRIVER
475 :MAINDEC-11-
476 :7 FEB 77 UPDATE: 18-APR-78
477 :L.PRUCHA V: 3
478 :THIS SOURCE TAKEN FROM BRUT2.LSI
479 :ENABLE ABS.AMA
480 :MCALL ..V2....REGDEF
481
482 .=0
483 000000 000000 .WORD 0
484 000002 000000 .WORD 0
485 000004 000000 .WORD 0
486 000006 000000 .WORD 0
487 000010 000000 .WORD 0
488 000012 000000 .WORD 0
489 000014 000000 .WORD 0
490 000016 000000 .WORD 0
491 000020 000000 .WORD 0
492 000022 000000 .WORD 0
493
494 .=200
495 000200 000167 000574 JMP START
496
497 ;***** RX INTERRUPT VECTOR *****
498
499 .=264 ;RX INTERRUPT HANDLER ADDRESS
500 000264 003452 RXINTR ;IF INTERRUPTS ARE USED AND VECTOR ADDRESS
501 000266 000340 340 ;IS NOT 264, THEN THIS MUST BE MODIFIED.
502
503 .=600
504 ;***** RX01,2,X REGISTERS *****
505
506 000600 177170 RXCS: 177170
507 000602 177172 RXDB: 177172
508
509 ;***** PROCESSOR ADDRESSES *****
510
511 000604 177776 PSW: 177776
512 000606 000760 SWR: 760
513
514 ;***** CONTSTANTS *****
515
516 000610 177570 HDSWR: 177570 ;HARDWARE SWITCH REG ADDRESS
517 000612 000760 SFSWR: 760 ;SOFTWARE SWITCH REG ADDRESS
518 000614 000004 BTRP: 4 ;BUS TRAP ADDRESS
519 000616 000006 BTRP2: 6 ;BUS TRAP PRI LEVEL

```

521 000700
 522
 523
 524 000700 000000
 525 000702 000001
 526 000704 100000
 527 000706 000001
 528 000710 000001
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 550
 551
 552 000712 000000
 553 000714 000000
 554 000716 000000
 555 000720 000000
 556 000722 000000
 557 000724 000000
 558 000726 000000
 559 000730 000000
 560 000732 000000
 561 000734 000000
 562 000736 000000
 563 000740 000000
 564 000742 000000
 565 000744 000000
 566 000746 000000
 567 000750 000000
 568
 569 004000
 570 005000
 571 003700
 572 004400
 573
 574 000760 000760
 575 000760 000000
 576

:=700
 :***** SET PARAMETERS DESIRED FOR UNIT UNDER TEST *****

SETCS: 0 ;SET RXCS-USED TO SET: DRV#,SIDE#,DENSITY BIT, INTERRUPT
 WRDCNT: 1 ;WORD COUNT-FOR FILL OR EMPTY BUFFER OPERATIONS
 RDYDLY: 100000 ;READY DELAY : TIME OUT
 RDYDX: 1 ;READY MULTIPLIER : DELAYS
 OPNUM: 1 ;NUMBER OF OPERATIONS (1 TO 15 DECIMAL)

***** FUNCTION TABLE *****

ENTER FUNCTIONS IN SEQUENCE DESIRED. MUST HAVE AT LEAST 1 FUNCTION ENTERED, AND MAY HAVE UP TO 16 OPERATIONS. SET THE OPERATION COUNTER (OPNUM) ABOVE EQUAL TO THE NUMBER OF FUNCTIONS IN THE SEQUENCE.

CODE	FUNCTION
0	= FILL BUFFER
1	= EMPTY BUFFER
2	= WRITE SECTOR
3	= READ SECTOR
4	= SET DENSITY (RX02,XX) ** BE CAREFUL TAKES 15 SECONDS **
5	= READ MAINTENANCE STATUS ** MAINTENANCE MODE **
6	= WRITE SECTOR WITH DELETED DATA
7	= READ ERROR CODE

!--< FILL WITH SEQUENCE OF FUNCTIONS

FUNTAB:	ETAD	XETAD (RX02,XX ONLY)
0	3700	4400
0	3702	4420
0	3704	4440
0	3706	4460
0	3710	4500
0	3712	4520
0	3714	4540
0	3716	4560
0	3720	4600
0	3722	4620
0	3724	4640
0	3726	4660
0	3730	4700
0	3732	4720
0	3734	4740
0	3736	4760

INTAB=4000 ;ADDRESS OF INPUT TABLE, RX TO TABLE
 OUTAB=5000 ;ADDRESS OF OUTPUT TABLE, TABLE TO RX
 ETBA=3700 ;ADDRESS OF ERROR TABLE (16 WORDS)
 XETBA=4400 ;ADDRESS OF EXTENDED ERROR CODE TABLE

:=760
 SOFTSR: 0 ;SOFTWARE SWITCH REG

```

578          001000          . =1000
579          : START OF PROGRAM
580          .SBTTL  MODULE 0.0 - CONTROL
581          ;-----
582
583 001000 012706 000500      START:  MOV   #500,SP          ;INITIALIZE STACK PIONTER
584 001004 012777 001042 177602      MOV   #SSWTRP,@BTRP      ;SET BUS TRAP FOR SWITCHLESS ADR TRAP
585 001012 012777 000200 177576      MOV   #200,@BTRP2      ;SET PROCESSOR PRI FOR TRAP
586 001020 005777 177564          TST   @HDSWR           ;CAUSE SWITCHLESS PROCESSOR TRAP
587 001024 016767 177560 177554      MOV   HDSWR,SWR        ;SET UP FOR HARDWARE SWITCH REG.
588 001032 016767 177552 002272      MOV   HDSWR,DISPLY     ;SET DISPLAY REG
589 001040 000403          BR    START0          ;NO SWITCHLESS PROCESSOR TRAP BRANCH
590 001042 016767 177544 177536      SSWTRP: MOV  SFSWR,SWR   ;SET UP SOFTWARE SWITCH REG.
591 001050 012777 001074 177536      START0: MOV  #LSITRP,@BTRP ;SET BUS TRAP FOR LSI TRAP
592 001056 012777 000200 177532      MOV   #200,@BTRP2      ;SET PROCESS
593 001064 012777 000340 177512      MOV   #340,@PSW        ;CAUSE LSI TRAP
594 001072 000405          BR    START1          ;BR IF NO TRAP
595 001074 012767 000001 000740      LSITRP: MOV  #1,LSIFLG  ;SET SWITCHLESS PROCESSOR FLAG
596 001102 106427 000200          MTPS  #200           ;SET LSI PRIORITY
597 001106 005077 177502          START1: CLR  @BTRP      ;RESET BUS TRAP
598 001112 005077 177500          CLR  @BTRP2          ;RESET BUS TRAP
599 001116 012767 000001 000642      MOV   #1,TR           ;INITIALIZE INPUT TYPE
600 001124 012767 000000 000632      MOV   #0,TE           ;INITIALIZE TABLE ENTRY
601 001132 012767 000000 000660      MOV   #0,DNWD         ;SET DONE BIT TO ZERO
602 001140 016767 177434 000636      MOV   CSAD,ADOT       ;ADDRESS OF OUTPUT WORD (PASS TO 3.0)
603 001146 016767 000652 000632      MOV   INTWD,WDOT      ;INITIALIZE WORD (PASS TO 3.0)
604 001154 004767 001076          JSR   PC,OUTPUT        ;INITIALIZE DEVICE DO 3.0
605 001160 012767 000040 000632      MOV   #40,DNWD        ;SET DONE BIT TO ONE
606 001166 016767 177406 000576      MOV   CSAD,ADIN       ;ADDRESS OF INPUT WORD (PASS TO 1.0)
607 001174 016767 000566 000574      MOV   TR,TYIN         ;INPUT TYPE (PASS TO 1.0)
608 001202 004767 000640          JSR   PC,INPUT         ;INPUT DEVICE CS REG DO 1.0
609 001206 016767 000562 000612      MOV   WDIN,CSWD       ;SAVE DEVICE CS REG (FROM 1.0)
610 001214 016767 177366 000550      DOO:  MOV   SWR,ADIN    ;ADDRESS OF WORD INPUT (PASS TO 1.0)
611 001222 012767 000000 000570      MOV   #0,DNWD         ;SET DONE BIT TO ZERO (PASS TO 1.0)
612 001230 012767 000001 000530      MOV   #1,TR           ;SET INPUT TYPE (PASS TO 1.0)
613 001236 016767 000524 000532      MOV   TR,TYIN         ;INPUT TYPE (PASS TO 1.0)
614 001244 004767 000576          JSR   PC,INPUT         ;INPUT SWITCH REG DO 1.0
615 001250 016767 000520 000552      MOV   WDIN,SWWD       ;SAVE SWITCH REG (FROM 1.0)
616 001256 016767 000502 000514      MOV   TE,TABENT       ;TABLE ENTRY (PASS TO 2.0)
617 001264 016767 177410 000524      MOV   SETFW,STWD      ;SET FUNCTION WORD (PASS TO 2.0)
618 001272 004767 000706          JSR   PC,PROCES        ;PROCESS TABLE ENTRY DO 2.0
619 001276 016767 000500 000456      MOV   FNWD,FW         ;SAVE FUNCTION WORD (FROM 2.0)
620 001304 032767 020000 000516      IFA0: BIT   #20000,SWWD ;IF SR BIT#13
621 001312 001413          BEQ   EA10           ;EQUALS ONE, THEN
622 001314 012767 000040 000476      MOV   #40,DNWD        ;SET DONE BIT (PASS TO 3.0)
623 001322 016767 177252 000454      MOV   CSAD,ADOT       ;AND SET ADDRESS OF OUTPUT (PASS TO 3.0)
624 001330 016767 000470 000450      MOV   INTWD,WDOT      ;AND SET INITIALIZE WORD (PASS TO 3.0)
625 001336 004767 000714          JSR   PC,OUTPUT        ;AND INITIALIZE DEVICE DO 3.0
626 001342 012767 000040 000450      EA10: MOV   #40,DNWD      ;SET DONE BIT TO ONE (PASS TO 3.0)
627 001350 016767 000410 000460      MOV   TE,TBEN         ;TABLE ENTRY (PASS TO 3.0)
628 001356 016767 000400 000422      MOV   FW,WDOT         ;FUNCTION WORD FOR OUTPUT (PASS TO 3.0)
629 001364 016767 000436 000376      MOV   CSWD,STATWD     ;DEVICE STATUS WORD (PASS TO 3.0)
630 001372 016767 177304 000410      MOV   WC,WDC          ;BUFFER WORD COUNT (RX02) (PASS TO 3.0)
631 001400 016767 000424 000404      MOV   SWWD,TASA       ;TA AND SA (IN SWITCH WORD) (PASS TO 3.0)
632 001406 004767 000644          JSR   PC,OUTPUT        ;OUTPUT FUNCTION WORD DO 3.0
633 001412 016701 000344          MOV   FW,R1           ;MOVE FUNCTION WORD TO R1
  
```

```

634 001416 042701 177761          BIC      #177761,R1      ;MASK ALL BUT FUNCTION
635 001422 022701 000002  IFB0:  CMP      #2,R1      ;IF FUNCTION IS (FW BITS#3,2,1)
636 001426 001023          BNE      ELB0         ;'EMPTY BUFFER' ( 0,0,1)
637 001430 016701 000372          MOV      CSWD,R1     ;THEN MOVE CS WORD TO R1
638 001434 032701 004000  IFCO:  BIT      #4000,R1  ;IF DEVICE IS
639 001440 001016          BNE      ELB0         ;RX01 (RXCS BIT#11=0)
640 001442 012767 000064 000316  MOV      #64,TR      ;THEN SET TYPE TRANSFER
641 001450 016767 000312 000320  MOV      TR,TYIN     ;INPUT TRANSFER (PASS TO 1.0)
642 001456 012767 000000 000334  MOV      #0,DNWD     ;SET DONE BIT TO ZERO (PASS TO 1.0)
643 001464 016767 177112 000300  MOV      DBAD,ADIN   ;ADDRESS OF INPUT (PASS TO 1.0)
644 001472 004767 000350          JSR      PC,INPUT    ;INPUT RX01 'EMPTY BUFF' DO 1.0
645 001476 016767 000262 000326  ELB0:  MOV      TE,MULTE   ;SET UP TABLE ENTRY
646 001504 006367 000322          ASL      MULTE       ;MULTIPLY TABLE ENTRY (*2)
647 001510 012767 003700 000316  MOV      #ETBA,ETAD  ;SET UP ERROR TABLE BASE ADDRESS
648 001516 066767 000310 000310  ADD      MULTE,ETAD  ;CALCULATE ERROR TABLE ADDRESS
649 001524 012767 000040 000270  MOV      #40,RDYWD   ;SET DONE BIT TEST (PASS TO 1/3.1)
650 001532 004767 001512          JSR      PC,DELAY    ;DELAY FOR DONE DO 1/3.1
651 001536 032767 000001 001344  IFD0:  BIT      #1,ERSTAT  ;IF ERROR STATUS
652 001544 001401          BEQ      ELDO        ;EQUALS 1 THEN
653 001546 000417          BR       EID0        ;PROCEED TO END IF 'D'
654 001550 012767 000001 000220  ELDO:  MOV      #1,TYIN   ;SET INPUT TYPE -SINGLE WD (PASS TO 1.0)
655 001556 012767 000040 000234  MOV      #40,DNWD    ;SET DONE BIT -COMPARE WD (PASS TO 1.0)
656 001564 016767 177010 000200  MOV      CSAD,ADIN   ;SET ADDRESS OF INPUT WD (PASS TO 1.0)
657 001572 004767 000250          JSR      PC,INPUT    ;GET DEVICE STATUS WD DO 1.0
658 001576 032767 100000 000170  IFE0:  BIT      #100000,WDIN ;IF DEVICE ERROR BIT (RXCS BIT#15=1)
659 001604 001422          BEQ      ELE0        ;EQUALS 1
660 001606 012767 000001 000162  EID0:  MOV      #1,TYIN   ;SET INPUT TYPE -SINGLE WD (PASS TO 1.0)
661 001614 016767 176762 000150  MOV      DBAD,ADIN   ;SET ADDRESS OF INPUT WD (PASS TO 1.0)
662 001622 012767 000040 000170  MOV      #40,DNWD    ;SET DONE BIT -COMPARE WD (PASS TO 1.0)
663 001630 004767 000212          JSR      PC,INPUT    ;GET DEVICE ERROR WD (RXES) DO 1.0
664 001634 016767 000134 000176  MOV      WDIN,ERWD   ;SAVE ERROR WD
665 001642 016777 000172 000164  MOV      ERWD,@ETAD  ;STORE ERROR WORD IN TABLE
666 001650 000405          BR       EIE0        ;PROCEED TO END IF 'E'
667 001652 012777 000000 000154  ELE0:  MOV      #0,@ETAD   ;ZERO ERROR TABLE LOCATION
668 001660 005067 001224          CLR      ERSTAT     ;CLEAR ERROR STATUS (SET BY 3.0)
669 001664 005267 000074          INC      TE         ;INCREMENT TABLE ENTRY
670 001670 032767 100000 000132  IFF0:  BIT      #100000,SWWD ;IF SR BIT#15 IS
671 001676 001411          BEQ      IFH0        ;EQUAL TO ONE
672 001700 000000          HALT              ;THEN HALT
673 001702 026767 177002 000054  IFG0:  CMP      TL,TE     ;IF (TL-TE)
674 001710 001003          BNE      EIG0        ;TE=TL
675 001712 012767 000000 000044  MOV      #0,TE      ;THEN SET TE=0
676 001720 000415          BR       EIF0        ;IF (TL-TE)
677 001722 026767 176762 000034  EIG0:  CMP      TL,TE     ;TE=TL
678 001730 001010          BNE      EIHO        ;THEN IF SR BIT#14 IS
679 001732 032767 040000 000070  IFI0:  BIT      #40000,SWWD ;EQUAL TO ONE
680 001740 001401          BEQ      EIIO        ;THEN HALT
681 001742 000000          HALT              ;SET TABLE ENTRY TO ZERO
682 001744 012767 000000 000012  EIIO:  MOV      #0,TE
683 001752 000240          EIHO:  NOP
684 001754 000240          EIF0:  NOP
685 001756 000167 177232          JMP      DOO        ;GET NEXT TABLE ENTRY
686

```

```

688      ;MODULE 0.0 - REGISTERS AND PARAMETERS -----
689      SWAD=SWR          ;EXTERNAL, SWITCH REG ADDRESS
690      CSAD=RXCS        ;EXTERNAL, RXCS ADDRESS
691      DBAD=RXDB        ;EXTERNAL, RXDB ADDRESS
692      TL=OPNUM         ;EXTERNAL, SET FUNCTION TABLE LENGTH
693      WC=WRDCNT        ;EXTERNAL, WORD COUNT
694      SETFW=SETCS      ;EXTERNAL, SET PART OF FUNCTION WORD
695      FW: 0            ;INTERNAL, FUNCTION WORD
696      TE: 0            ;INTERNAL, TABLE ENTRY
697      TR: 0            ;INTERNAL, TRANSFER TYPE
698      STATWD: 0        ;MODULE 3.0 DEVICE STATUS WORD
699      ADIN: 0          ;MODULE 1.0 INPUT ADDRESS
700      WDIN: 0          ;MODULE 1.0 INPUT WORD
701      TYIN: 0          ;MODULE 1.0 TYPE TRANSFER
702      TABENT: 0        ;MODULE 2.0 TABLE ENTRY
703      FNWD: 0          ;MODULE 2.0 FUNCTION WORD
704      ADOT: 0          ;MODULE 3.0 OUTPUT ADDRESS
705      WDOT: 0          ;MODULE 3.0 OUTPUT WORD
706      WDCT: 0          ;MODULE 3.0 WORD COUNT (RX02)
707      TASA: 0          ;MODULE 3.0 TRACK AND SECTOR ADDRESS
708      DATAD: 0         ;MODULE 3.0 DATA ADDRESS
709      STWD: 0          ;MODULE 2.0 PRESET FUNCTION WORD
710      DNWD: 0          ;MODULE 0.0 DONE READY TEST WORD
711      RDYWD: 0         ;MODULE 1/3.1 READY WORD
712      INTWD: 40000     ;INTERNAL, INITIALIZE DEVICE WORD
713      CSWD: 0          ;INTERNAL, DEVICE CS REG
714      SWWD: 0          ;INTERNAL, SWITCH REG WORD
715      MULTE: 0         ;INTERNAL, MULTIPLY TABLE ENTRY PARAM.
716      ETAD: 0          ;INTERNAL, ADDRESS OF ERROR WORD
717      TBEN: 0          ;MODULE 3.0 CURRENT TABLE ENTRY
718      ERWD: 0          ;INTERNAL, TEMP STORAGE ERROR WORD
719      LSIFLG: 0        ;EXTERNAL, LSI PROCESSOR/SWITCHLESS PROCESSOR FLAG
720      INTRFG: 0        ;EXTERNAL, INTERRUPT FLAG
721
722
723      ;MODULE 0.0 ----- END MODULE -----

```


725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762

.SBTTL MODULE 1.0 - INPUT

```

-----
INPUT:  NOP
IFA1:  BIT    #1, TYIN      ;IF WORD TRANSFER
      BEQ    ELA1         ;EQUALS ONE
      MOV    DNWD, RDYWD   ;SET READY WORD (PASS TO 1/3.1)
      JSR    PC, DELAY    ;DELAY FOR READY DO 1/3.1
      NOP
      MOV    @ADIN, WDIN  ;THEN TRANSFER WORD (PASS TO 0.0)
      NOP
      BR    EIA1         ;BRANCH TO ENDIF 'A'
ELA1:  MOV    #0, BYCNT   ;INITIALIZE BYTE COUNT
BDA1:  MOV    BYCNT, R1    ;MOVE BYTE COUNT
      ADD    #BAINTB, R1  ;ADD DATA BASE ADDRESS TO BYTE COUNT
      MOV    R1, DAINAD   ;MOVE RESULT TO DATA ADDRESS
      MOV    INTR, RDYWD  ;SET READY WORD (PASS TO 1/3.1)
      JSR    PC, DELAY    ;DELAY FOR READY DO 1/3.1
      MOVB   @ADIN, @DAINAD ;MOV DATA BYTE TO INPUT DATA TABLE
      INC    BYCNT       ;INCREMENT BYTE COUNT
      NOP
      NOP
      BIT    #200, BYCNT  ;TEST BYCNT
DUA3:  BEQ    BDA1       ;DUNTIL BYCNT=128 BYTES
EIA1:  NOP
      NOP
      RTS    PC          ;RETURN TO MOD 0.0
  
```

;MODULE 1.0 REGISTERS -----

```

BYCNT:  0      ;INTERNAL, BYTE COUNTER
DAINAD: 0      ;INTERNAL, CURRENT ADDRESS DATA INPUT TABLE
INTR:   200   ;INTERNAL, TRANSFER READY, INPUT
BAINTB=INTAB ;EXTERNAL, INPUT DATA TABLE BEGIN ADDRESS
;MODULE 1.0 ----- END MODULE -----
  
```

764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785

002204 000240
002206 006167 177566
002212 016705 177562
002216 012704 000712
002222 060405
002224 010567 000024
002230 011501
002232 000261
002234 006101
002236 016702 177554
002242 050102
002244 010267 177532
002250 000240
002252 000207

000712
000000

```
.SBTTL MODULE 2.0 - PROCESS  
-----  
PROCES: NOP  
ROL TABENT ;DOUBLE ENTRY FOR TABLE ADDRESS  
MOV TABENT,R5 ;ENTRY FUNCTION CODE TABLE  
MOV #TBA,R4 ;BASE ADDRESS FUNCTION CODE TABLE  
ADD R4,R5 ;FORM ADDRESS OR FUNCTION CODE  
MOV R5,FUNCAD ;SAVE ADDRESS OF FUNCTION CODE  
MOV (R5),R1 ;GET FUNCTION CODE  
SEC ;SET CARRY BIT  
ROL R1 ;FORMAT FUNCTION CODE + GO BIT  
MOV STWD,R2 ;GET SET FUNCTION WORD  
BIS R1,R2 ;MASK FUNCTION CODE ONTO SET FUNCTION WORD  
MOV R2, FNWD ;FUNCTION WORD (PASS TO 0.0)  
NOP  
RTS PC ;RETURN TO MOD 0.0  
;MODULE 2.0 REGISTERS -----  
TBA=FUNTAB ;BEGIN ADDRESS FUNCTION CODE TABLE  
FUNCAD: 0 ;CURRENT FUNCTION CODE ADDRESS  
;MODULE 2.0 ----- END MODULE -----
```

```

787          .SBTTL  MODULE 3.0 - OUTPUT
788          -----
789
790 002256 000240          OUTPUT: NOP
791 002260 016767 177534 177534      MOV    DNWD,RDYWD      ;READY TEST WD      (PASS TO 3.1)
792 002266 016767 177514 000604      MOV    WDOT,WRDS      ;WORD FOR OUTPUT    (PASS TO 3.1)
793 002274 016767 176300 000600      MOV    CSADR,ADRS     ;ADDRESS OF OUTPUT  (PASS TO 3.1)
794 002302 004767 000614              JSR    PC,OUTSWD      ;OUTPUT FUNCTION WD (FW) DO 3.1
795 002306 032767 040000 177472  IFA3: BIT    #40000,WDOT    ;IF FUNCTION IS
796 002314 001001              BNE    ELA3          ;NOT AN 'INITIALIZE' (FW BIT#14=0)
797 002316 000402              BR     THA3          ;THEN 'A'
798 002320 000167 000520              ELA3: JMP    END3      ;ENDIF 'A' -DONE
799 002324 032767 000010 177454  THA3: BIT    #10,WDOT    ;THEN, IF FUNCTION IS
800 002332 001073              BNE    IFC3          ;'READ,WRITE,FILL,EMPTY' (FW BIT#3=0)
801 002334 032767 000004 177444  IFH3: BIT    #4,WDOT     ;AND THEN IF FUNCTION IS
802 002342 001077              BNE    ELH3          ;'EMPTY,FILL' (FW BIT#2=0)
803 002344 032767 004000 177416  :FI3: BIT    #4000,STATWD ;THEN IF DEVICE IS
804 002352 001443              BEQ    IFJ3          ;RX02 (RXCS BIT#11=1)
805 002354 016767 000516 177440      MOV    OTTR,RDYWD     ;THEN SET OUTPUT READY TEST WORD (PASS TO 3.1)
806 002362 016767 177422 000510      MOV    WDCT,WRDS      ;AND SET WORD FOR OUTPUT (PASS TO 3.1)
807 002370 016767 176206 000504      MOV    DBADR,ADRS     ;AND SET ADDRESS OF OUTPUT (PASS TO 3.1)
808 002376 004767 000520              JSR    PC,OUTSWD      ;OUTPUT WORD COUNT WORD DO 3.1
809 002402 032767 000002 177376  IFK3: BIT    #2,WDOT     ;IF FUNCTION IS
810 002410 001004              BNE    ELK3          ;'FILL BUFFER' (FW BIT#1=0)
811 002412 012767 005000 000474      MOV    #BAOUTB,BA     ;SET BASE ADDRESS FOR FILL
812 002420 000403              BR     EIK3          ;BRANCH TO ENDIF 'K'
813 002422 012767 004000 000464  ELK3: MOV    #BAINTB,BA  ;SET BASE ADDRESS FOR EMPTY
814 002430 016767 000442 177364  EIK3: MOV    OTTR,RDYWD ;SET OUTPUT READY TEST WORD (PASS TO 3.1)
815 002436 016767 000452 000434      MOV    BA,WRDS        ;WORD FOR OUTPUT (PASS TO 3.1)
816 002444 016767 176132 000430      MOV    DBADR,ADRS     ;ADDRESS OF OUTPUT (PASS TO 3.1)
817 002452 004767 000444              JSR    PC,OUTSWD      ;OUTPUT BASE ADDRESS WORD DO 3.1
818 002456 000167 000362              JMP    END3          ;DONE
819 002462 032767 000002 177316  IFJ3: BIT    #2,WDOT     ;IF FUNCTION IS
820 002470 001013              BNE    EI13          ;'FILL BUFFER -RX01' (FW BIT#1=0)
821 002472 016767 000400 177322      MOV    OTTR,RDYWD     ;THEN SET OUTPUT READY TEST WORD (PASS TO 3.2)
822 002500 016767 002274 000376      MOV    BAOUTB,WRDM    ;WORD TABLE FOR OUTPUT (PASS TO 3.2)
823 002506 016767 176070 000372      MOV    DBAD,ADRM      ;ADDRESS OF OUTPUT (PASS TO 3.2)
824 002514 004767 000432              JSR    PC,OUTMWD      ;OUTPUT WORD TABLE DO 3.2
825 002520 000460              BR     EI13          ;BRANCH TO ENDIF 'H'
826 002522 032767 000004 177256  IFC3: BIT    #4,WDOT     ;IF FUNCTION WORD IS
827 002530 001516              BEQ    IFE3          ;'WRITE D.D.' OR 'READ E.C.'(FW BIT#2=1)
828 002532 032767 000002 177246  IFD3: BIT    #2,WDOT     ;THEN, IF FUNCTION IS
829 002540 001051              BNE    IFF3          ;'WRITE D.D.', THEN (FW BIT#1=0)
830 002542 016767 000330 177252  ELH3: MOV    OTTR,RDYWD ;SET OUTPUT READY TEST WORD (PASS TO 3.1)
831 002550 016767 177236 000322      MOV    TASA,WRDS      ;MOVE TRACK AND SECTOR ADDRESS (PASS TO 3.1)
832 002556 042767 177700 000314      BIC    #177700,WRDS   ;FORMAT TO SECTOR ADDRESS (PASS TO 3.1)
833 002564 016767 176012 000310      MOV    DBAD,ADRS     ;ADDRESS OF OUTPUT (PASS TO 3.1)
834 002572 004767 000324              JSR    PC,OUTSWD      ;OUTPUT SECTOR ADDRESS DO 3.1
835 002576 016767 177210 000274      MOV    TASA,WRDS      ;MOVE TRACK AND SECTOR ADDRESS
836 002604 006067 000270              ROR    WRDS          ;FORMAT
837 002610 006067 000264              ROR    WRDS          ;FORMAT
838 002614 006067 000260              ROR    WRDS          ;FORMAT
839 002620 006067 000254              ROR    WRDS          ;FORMAT
840 002624 006067 000250              ROR    WRDS          ;FORMAT
841 002630 006067 000244              ROR    WRDS          ;FORMAT
842 002634 042767 177600 000236      BIC    #177600,WRDS   ;FORMAT TRACK ADDRESS (PASS TO 3.1)

```

```

843 002642 016767 000230 177152      MOV      OTTR,RDYWD      ;SET OUTPUT READY TEST WORD (PASS TO 3.1)
844 002650 016767 175726 000224      MOV      DBADR,ADRS     ;ADDRESS OF OUTPUT          (PASS TO 3.1)
845 002656 004767 000240                JSR      PC,OUTSWD      ;OUTPUT TRACK ADDRESS      DO 3.1
846 002662 000470                BR       END3           ;ENDIF H -DONE
847 002664 032767 004000 177076  IFF3:    BIT      #4000,STATWD   ;IF DEVICE IS
848 002672 001441                BEQ      THE3           ;RX02, THEN          (RXCS BIT#11=1)
849 002674 016767 177136 000214      MOV      TBEN,MULT      ;SET UP TABLE ENTRY FOR MULT.
850 002702 006367 000210                ASL      MULT           ;-MULTIPLY
851 002706 006367 000204                ASL      MULT           ; TABLE ENTRY (TE)
852 002712 006367 000200                ASL      MULT           ; BY
853 002716 006367 000174                ASL      MULT           ; 20(8)
854 002722 012767 004400 000170      MOV      #XETBA,XETAD   ;SET UP TO CAL. ADDRESS
855 002730 066767 000162 000162      ADD      MULT,XETAD     ;EXTENDED ERROR TABLE ADDRESS
856 002736 016767 000134 177056      MOV      OTTR,RDYWD     ;SET READY WD TO TR MODE   (PASS TO 3.1)
857 002744 016767 000150 000126      MOV      XETAD,WRDS     ;EXT ERR. CODE TABLE ADD (PASS TO 3.1)
858 002752 016767 175624 000122      MOV      DBADR,ADRS    ;ADDRESS OF OUTPUT, RXDB  (PASS TO 3.1)
859 002760 004767 000136                JSR      PC,OUTSWD     ;O/P BASE ADD FOR ERR. CODE DO 3.1
860 002764 000427                BR       END3           ;DONE
861 002766 032767 000002 177012  IFE3:    BIT      #2,WDOT        ;IF FUNCTION IS
862 002774 001404                BEQ      ELE3           ;'READ STATUS'        (FW BIT#1=1)
863 002776 012767 000001 000104  THE3:    MOV      #1,ERSTAT     ;THEN-SET ERR STATUS FLAG (CLEAR BY 0.0)
864 003004 000417                BR       END3           ;DONE
865 003006 032767 004000 176754  ELE3:    BIT      #4000,STATWD   ;IF DEVICE IS
866 003014 001413                BEQ      END3           ;RX02, THEN          (RXCS BIT#11=1)
867 003016 016767 000054 176776      MOV      OTTR,RDYWD     ;SET OUTPUT READY TEST WD (PASS TO 3.1)
868 003024 016767 000062 000046      MOV      VALWD,WRDS     ;VALIDATION WORD        (PASS TO 3.1)
869 003032 016767 175544 000042      MOV      DBADR,ADRS    ;ADDRESS OF OUTPUT ,RXDB (PASS TO 3.1)
870 003040 004767 000056                JSR      PC,OUTSWD     ;OUTPUT VALIDATION WORD  DO 3.1
871 003044 032767 000100 176734  END3:    BIT      #100,WDOT      ;IF INTERRUPT BIT
872 003052 001403                BEQ      1$             ;IS SET, THEN
873 003054 004767 000254                CALL     WATCH          ;CALL MOD U.2 - WATCH DOG
874 003060 000405                BR       2$             ;BR TO END
875 003062 012767 000040 176732  1$:     MOV      #40,RDYWD      ;SET DELAY FOR DONE
876 003070 004767 000154                JSR      PC,DELAY       ;DELAY FOR DONE BIT
877 003074 000207                2$:     RTS      PC          ;RETURN TO MOD 0.0
878                                     ;MODULE 3.0 REGISTERS -----
879
880 003076 000200      OTTR:    200             ;MODULE 1/3.1 TRANSFER READY, OUTPUT
881 003100 000000      WRDS:    0              ;MODULE 3.1 OUTPUT WORD
882 003102 000000      ADRS:    0              ;MODULE 3.1 OUTPUT ADDRESS
883 003104 000000      WRDM:    0              ;MODULE 3.2 OUTPUT WORD
884 003106 000000      ADRM:    0              ;MODULE 3.2 OUTPUT ADDRESS
885 003110 000000      ERSTAT:  0              ;MODULE 0.0 ERR STATUS READ FLAG
886 003112 000111      VALWD:  111            ;EXTERNAL, VALIDATION WD (SET DENS.) ASCII 'I'
887 003114 000000      BA:      0              ;INTERNAL, TEMP BASE ADDRESS
888 003116 000000      MULT:    C              ;INTERNAL, MULTIPLY PARAMETER
889 003120 000000      XETAD:   0              ;MODULE 3.1 EXT. ERR. TABLE ADDR.
890                DBADR=DBADR          ;MODULE 0.0 RX DATA BUFFER ADDRESS
891                CSADR=CSADR         ;MODULE 0.0 RX CONT/STATUS ADDRESS
892                BAOUTB=OUTAB       ;EXTERNAL, BASE ADDRESS OUTPUT TABLE
893
;MODULE 3.0 ----- END MODULE -----

```

```
895      .SBTTL  MODULE 3.1 - OUTPUT SINGLE WORD
896      ;-----
897
898 003122 000240      OUTSWD: NOP
899 003124 016767 176672 176670      MOV      RDYWD,RDYWD      ;OUTPUT READY WORD      (PASS TO 1/3.1)
900 003132 004767 000112      JSR      PC,DELAY      ;DELAY FOR READY      DO 1/3.1
901 003136 000240      NOP
902 003140 016777 177734 177734      MOV      WRDS,@ADRS      ;MOVE WORD TO ADDRESS
903 003146 000240      NOP
904 003150 000207      RTS      PC      ;RETURN TO MOD 3.0
905      ;MODULE 3.1 ----- END MODULE -----
906
907      .SBTTL  MODULE 3.2 - OUTPUT MULTIPLE WORDS
908      ;-----
909
910 003152 000240      OUTMWD: NOP
911 003154 012767 000000 000062      MOV      #0,BTCNT      ;INITIALIZE BYTE COUNT
912 003162 016767 176634 176632      BDA32: MOV      RDYWD,RDYWD      ;OUTPUT READY WORD      (PASS TO 1/3.1)
913 003170 004767 000054      JSR      PC,DELAY      ;DELAY FOR READY      DO 1/3.1
914 003174 000240      NOP
915 003176 016701 000042      MOV      BTCNT,R1      ;MOVE BYTE COUNT
916 003202 062701 005000      ADD      #BAOUTB,R1      ;ADD DATA BASS ADDRESS TO BYTE COUNT
917 003206 010167 000034      MOV      R1,DAOTAD      ;MOV RESULT TO OUTPUT DATA ADDRESS
918 003212 117777 000030 177666      MOVVB   @DAOTAD,@ADRM      ;MOV DATA BYTE TO OUTPUT
919 003220 000240      NOP
920 003222 005267 000016      INC      BTCNT      ;INCREMENT BYTE COUNT
921 003226 000240      NOP
922 003230 032767 000200 000006      DUA32: BIT      #200,BTCNT      ;TEST BYCNT
923 003236 001751      BEQ      BDA32      ;DO UNTIL BTCNT=128 BYTES
924 003240 000240      NOP
925 003242 000207      RTS      PC      ;RETURN TO MOD 3.0
926      ;MODULE 3.2 REGISTERS -----
927
928 003244 000000      BTCNT: 0      ;INTERNAL, BYTE COUNTER
929      BAOUTB=OUTAB      ;EXTERNAL, BASE ADDRESS OF OUTPUT TABLE
930 003246 000000      DAOTAD: 0      ;INTERNAL, CURRENT ADDRESS IN OUTPUT TABLE
931      ;MODULE 3.2 ----- END MODULE -----
```

```

933          .SBTTL  MODULE 1/3.1 - DELAY
934          ;-----
935
936 003250 000240          DELAY:  NOP
937 003252 026727 176544 000000  IFA13:  CMP      RDYWD,#0      ;IF READY WORD
938 003260 001423          BEQ      EIA13      ;EQUALS ZERO THEN BRANCH TO ENDIF 'A'
939 003262 016704 175420          MOV      RYDX,R4      ;SET READY DELAY MULT
940 003266 016703 175412          BDA13:  MOV      RYDLY,R3      ;SET READY DELAY
941 003272 036777 176524 175300  BDB13:  BIT      RDYWD,@CSAD    ;IF READY
942 003300 001012          BNE      EIB13      ;EQUAL TO ONE THEN BRANCH TO ENDIF 'B'
943 003302 005303          DEC      R3          ;ELSE DECREMENT DELAY
944 003304 010377 000022          MOV      R3,@DISPLY  ;DISPLAY R3
945 003310 010377 000016          MOV      R3,@DISPLY  ;DISPLAY R3
946 003314 010377 000012          MOV      R3,@DISPLY  ;DISPLAY R3
947 003320 001364          BNE      BDB13      ;DUNTIL R3=0
948 003322 005304          DEC      R4          ;DECREMENT DELAY MULT.
949 003324 001360          BNE      BDA13      ;DUNTIL R4=0
950 003326 000240          EIB13:  NOP
951 003330 000207          EIA13:  RTS      PC      ;RETURN TO CALLING MODULE
952          ;MODULE 1/3.1 REGISTERS -----
953
954          RYDX=RDYDX      ;EXTERNAL, READY MULTIPLIER
955          RYDLY=RDYDLY    ;EXTERNAL, READY DELAY
956 003332 000762          DISPLY: 762          ;INTERNAL, ADDRESS OF LIGHTS
957          ;MODULE 1/3.1 ----- END MODULE -----
  
```

```

959          .SBTTL MOD U.2.1 - WATCH DOG TIMER
960          ;-----
961          ;
962 003334 000240          WATCH: NOP          ;
963 003336 012767 000000 000104      MOV      #PRO,NEWPRI      ;SET PROCESSOR PRI=0
964 003344 004767 000054          JSP      PC,SETPRI      ;SET PROCESSOR PRI
965 003350 016704 000044          MOV      DX,R4          ;SET DELAY MULT
966 003354 016703 000042      BDAU21: MOV     DLY,R3      ;SET DELAY
967 003360 005767 176460      BDBU21: TST     INTRFG     ;IF INTERRUPT FLAG
968 003364 001007          BNE      ENDU21         ;EQUALS ZERO, THEN
969 003366 005303          DEC      R3            ;DECREMENT DELAY CNT
970 003370 001373          BNE      BDBU21         ;IF IT EQUALS ZERO
971 003372 005304          DEC      R4            ;THEN DECREMENT DELAY MULT
972 003374 001367          BNE      BDAU21         ;IF IT EQUALS ZERO, THEN
973 003376 052767 000001 177504      BIS      #1,ERSTAT      ;SET ERROR
974 003404 012767 000340 000036      ENDU21: MOV     #PR7,NEWPRI ;SET PROCESSOR PRI=7
975 003412 004767 000006          JSR      PC,SETPRI      ;SET PROCESSOR PRI
976 003416 000207          RTS      PC            ;RETURN TO MOD 2.3.4
977          ;-----
978          PRO=0          ;PRIORITY 0
979          PR7=340        ;PRIORITY 7
980 003420 000010      DX: 10          ;DELAY MULT
981 003422 100000      DLY: 100000     ;DELAY
982          ;MOD U.2.1 ---- END MODULE ----
983          ;-----
984          .SBTTL SET PROCESSOR PRI
985          ;-----
986          ;
987 003424 005767 176412      SETPRI: TST     LSIFLG     ;IF PROCESSOR IS
988 003430 001403          BEQ      1$            ;LSI, THEN
989 003432 106467 000012          MTPS     NEWPRI         ;SET PROCESSOR PRI
990 003436 000403          BR      SETPIX         ;BR TO END
991 003440 016777 000004 175136      1$: MOV      NEWPRI,@PSW   ;SET PROCESSOR PRI
992 003446 000207      SETPIX: RTS      PC      ;RETURN
993          ;-----
994 003450 000000      NEWPRI: 0          ;NEW PROCESSOR PRIORITY
995          ;-----
996          ;
997          .SBTTL RX INTERRUPT HANDLER
998          ;-----
999          ;
1000 003452 000240      RXINTR: NOP          ;
1001 003454 005267 176364          INC      INTRFG         ;INCREMENT INTERRUPT FLAG
1002 003460 000240          NOP          ;
1003 003462 000002          RTI          ;RETURN TO PROGRAM
1004          ;-----

```

```

1006      .SBTTL DATA TABLES
1007      ;-----
1008      003700      .=3700      ;ERROR TABLE (CODE=5 OR RXES IF ERR OCCURRED)
1009      000020      .REPT 16.
1010      .WORD 0
1011      .ENDR
1012
1013      ;-----
1014      004000      .=4000      ;INPUT TABLE -- RX TO TABLE
1015      000400      .REPT 256.
1016      .BYTE 0
1017      .ENDR
1018
1019      ;-----
1020      004400      .=4400      ;EXTENDED ERROR CODE TABLE (CODE=7)
1021      000200      .REPT 128.
1022      .WORD 0
1023      .ENDR
1024
1025      ;-----
1026      005000      .=5000      ;OUTPUT TABLE -- TABLE TO RX
1027      000400      .REPT 256.
1028      .BYTE 377
1029      .ENDR
1030
1031      005400 000000  PATCH: 0      ;PATCH AREA
1032      006000      .=6000
1033      ;-----
1034
1035      ;----- BOOT RX01 -----
1036      006000 005000  BOOTRX: CLR R0
1037      006002 012701 177170  MOV #177170,R1 ;LOAD RXCS ADR
1038      006006 105711  1$: TSTB (R1) ;TEST RXCS
1039      006010 001776  BEQ 1$
1040      006012 012711 000003  MOV #3,(R1) ;SEND COMMAND
1041      006016 005711  2$: TST (R1) ;TEST RXCS
1042      006020 001776  BEQ 2$ ;WAIT FOR TR, DONE, OR ERROR
1043      006022 100405  BMI 3$ ;ERROR - HALT!
1044      006024 105711  TSTB (R1) ;SEE IF 'TR'
1045      006026 100004  BPL 4$ ;NOT 'TR', BUT 'DONE'
1046      006030 116120 000002  MOVB 2(R1),(R0)+ ;MOVE BYTE
1047      006034 000770  BR 2$ ;NEXT
1048      006036 000000  3$: HALT
1049      006040 005000  4$: CLR R0
1050      006042 000110  JMP (R0)
1051      006044 000000  HALT
1052      006046 000000  HALT
1053      006050 000000  HALT
1054      ;-----
1055
1056      000001      .END
1057

```


RXDB	000602	507#	691																
RXINTR	003452	500	1000#																
RYDLY =	000704	940	955#																
RYDX =	000706	939	954#																
SETCS	000700	524#	694																
SETFW =	000700	617	694#																
SETPIX	003446	990	992#																
SETPRI	003424	964	975	987#															
SFSWR	000612	517#	590																
SOF TSR	000760	575#																	
SSWTRP	001042	584	590#																
START	001000	495	583#																
START0	001050	589	591#																
START1	001106	594	597#																
STATWD	001770	629*	698#	803	847	865													
STWD	002016	617*	709#	776															
SWAD =	000606	689#																	
SWR	000606	512#	587*	590*	610	689													
SWWD	002030	615*	620	631	670	679	714#												
TABENT	002000	616*	702#	768*	769														
TASA	002012	631*	707#	831	835														
TBA =	000712	770	783#																
TBEN	002036	627*	717#	849															
TE	001764	600*	616	627	645	669*	673	675*	677	682*	696#								
THA3	002324	797	799#																
THE3	002776	848	863#																
TL =	000710	673	677	692#															
TR	001766	599*	607	612*	613	640*	641	697#											
TYIN	001776	607*	613*	641*	654*	660*	701#	731											
VALWD	003112	868	886#																
WATCH	003334	873*	962#																
WC =	000702	630	693#																
WDCT	002010	630*	706#	806															
WDIN	001774	609	615	658	664	700#	736*												
WDOT	002006	603*	624*	628*	705#	792	795	799	801	809	819	826	828	861					
		871																	
WRDCNT	000702	525#	693																
WRDM	003104	822*	883#																
WRDS	003100	792*	806*	815*	831*	832*	835*	836*	837*	838*	839*	840*	841*	842*					
		857*	868*	881#	902														
XE TAD	003120	854*	855*	857	889#														
XE TBA =	004400	572#	854																
	= 006052	482#	494#	499#	503#	521#	574#	578#	1008#	1014#	1020#	1026#	1032#						

.REGDE 480#
..V2.. 480#

. ABS. 006052 000

ERRORS DETECTED: 0

CZRXC.A,CZRXC.A/CRF=CZRXC.A.P11
RUN-TIME: 4 2 .3 SECONDS
RUN-TIME RATIO: 10/7=1.3
CORE USED: 9K (18 PAGES)