# DV11

BSC TST & ROM INST EX

CZDVAC0

AH-8729C-MC

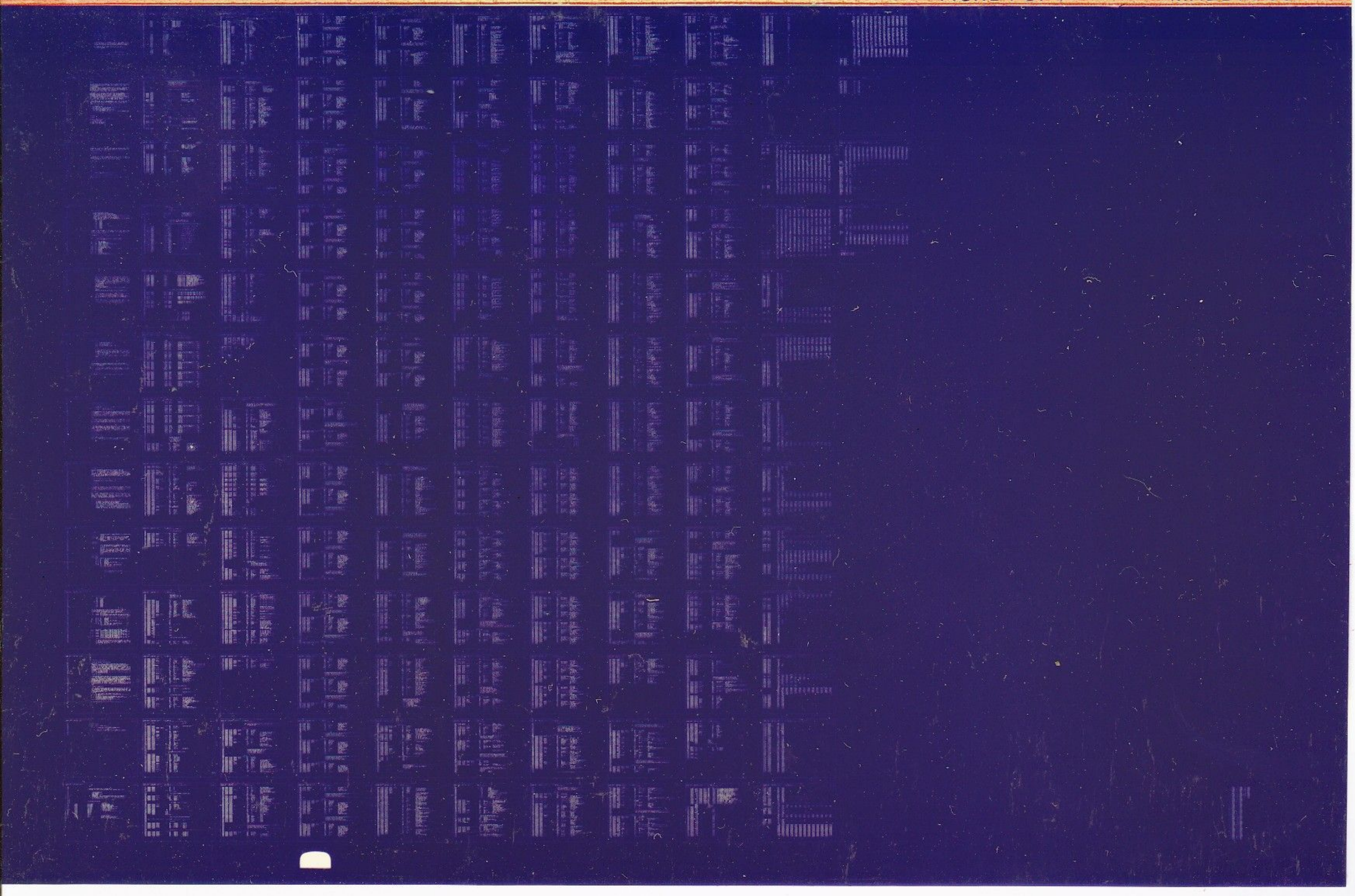


FICHE 1 OF 1

JAN 1980

digital

MADE IN USA



```
.REM $
```

## IDENTIFICATION

PRODUCT CODE: AC-8728C-MC

PRODUCT NAME: CZDVACO BSC TST & ROM INST EX

PRODUCT DATE: OCTOBER, 1979

MAINTAINER: DIAGNOSTIC ENGINEERING

1. ABSTRACT

THE FUNCTION OF THE DV11 DIAGNOSTICS ARE TO VERIFY THAT THE OPTION OPERATES ACCORDING TO SPECIFICATIONS. THE DIAGNOSTICS VERFIY THAT THERE ARE NO MALFUNCTIONS AND THE ALL OPERATIONS OF THE DV11 ARE CORRECT IN ITS ENVIRONMENT.
PARAMETERS MAY BE SET TO ALERT DIAGNOSTICS AS TO THE DV11 CONFIGURATION BY USING THE "TRIAL" PROGRAM (CZDVE SA:210). ALL QUESTIONS SHOULD BE ANSWERED AND THEN EACH DIAGNOSTIC WILL "OVERLAY" THESE PARAMETERS WHICH ARE STORED IN THE "STATUS TABLE" (SEE SECTION 8.4A). THE ALTERNATIVE TO "TRIAL" PROGRAM IS "AUTO SIZING" (SEE SECTION 8.5).

CZDVA DOES R/W TESTS ON BOTH PRIMARY REGISTERS AND ALL SECONDARY REGISTERS. TESTS ARE MADE TO VERIFY THAT NO INTER-ACTION BETWEEN SECONDARY REGISTERS OF ANY LINES EXISTS. CZDVA ALSO EXERCISES ALL MICRO-CODE INSTRUCTIONS AND VERIFIES INTERNAL REGISTERS USED BY THE MICRO PROCESSOR. INTERRUPTS AND NPR'S ARE ALSO TESTED IN THIS DIAGNOSTIC.
NOTE: FOR EASE OF DIAGNOSIS; ALL (4) LINE CARDS MAY BE PULLED OUT OF THE SYSTEM UNIT AS MAY THE TWO MODEM CONTROL MODULES. ALSO THE DIAGNOSTIC IN NO WAY READS OR USES THE ROMS THAT ARE IN THE DV11.

CURRENTLY THERE ARE SIX OFF LINE DIAGNOSTICS THAT ARE TO BE RUN IN SEQUENCE TO INSURE THAT IF AN ERROR SHOULD OCCUR IT WILL BE DETECTED AT AN EARLY STAGE AND INSURING THAT DIAGNOSIS OF ERROR WILL BE IMMEDIATE TO PROBLEM
NOTE: ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.

THE SIX DIAGNOSTICS ARE:
1. CZDVA [REV] BASIC R/W TEST AND ROM INSTRUCTION EXERCISER.
2. CZDVB [REV] STATIC LINE CARD TESTS.
3. CZDVC [REV] 'FREE RUNNING' ROM TESTS PART 1.
4. CZDVD [REV] 'FREE RUNNING' ROM TESTS PART 2.
5. CZDVE [REV] MODEM CONTROL AND CABLE TESTS PLUS MANUAL PARAMETER INPUT. [TRIAL PROGRAM]
6. CZDVF [REV] ASYNCHRONOUS LINE CARD TESTS.

2. REQUIREMENTS

2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (WITH MINIMUM 8K MEMORY)
ASR 33 (OR EQUILIVALENT)
DV11-AA MUX CNTRL UNIT
AT LEAST ONE OF THE FOLLOWING
DV11-BA 8 LINE SYNC MODULES
DV11-BB 8 LINE ASYNC MODULES
DV11-BC 4 SYNC LINES, 4 ASYNC LINES

```
                        2.2     STORAGE

                                PROGRAM WILL USE ALL 8K OF MEMORY EXCEPT WHERE ABL AND BOOTSTRAP  LOADER
                                RESIDE.   LOCATION  1500  THRU 1736 ARE ESPECIALLY TO BE NOTED AND TO BE
                                UNTOUCHED BY OPERATOR AFTER DV11 TRIAL PROGRAM HAS  BEEN  EXECUTED;   OR
                                AFTER THE 'AUTO SIZING' HAS BEEN DONE.

                        3.      LOADING PROCEEDURE

                        3.1     METHOD

                                ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED  USING  THE  ABSOLUTE
                                LOADER.  NOTE:  IF   THE  DIAGNOSTICS  ARE  ON  A  MEDIA  SUCH  AS  DISK
                                ,MAGTAPE,DECTAPE, OR CASSETTE;  FOLLOW  INSTRUCTIONS  FOR  THE  MONITOR
                                WHICH HAS BEEN PROVIDED ON THAT SPECIFIC MEDIA.

                                ABSOLUTE LOADER STARTING ADDRESS *500

                                MEMORY  * SIZE

                                4K      17
                                8K      37
                                12K     57
                                16K     77
                                20K     117
                                24K     137
                                28K     157


                        3.1.1   PLACE ADDRESS OF ABS LOADER INTO SWITCH REGISTER.
                                        (ALSO PLACE 'HALT' SW UP)

                        3.1.2   DEPRESS 'LOAD ADDRESS' KEY ON CONSOLE AND RELEASE.

                        3.1.3   DEPRESS 'START KEY' ON  CONSOLE  AND  RELEASE  (PROGRAM  SHOULD  NOW  BE
                                LOADING INTO CPU)
```

4. STARTING PROCEEDURE

A. SET SWITCH REGISTER TO 000200
B. DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
C. SET SWR TO ZERO FOR 'AUTO SIZING' OR LEAVE
LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS SET UP BY DV11 TRIAL PROGRAM OR A PREVIOUSLY RUN DV11 DIAGNOSTIC THAT USED THE 'AUTO SIZING'.(SECTION 7.2 AND 8.4,8.5 MAY BE HELPFUL)
D. DEPRESS 'START KEY' AND RELEASE ; THE PRROGRAM WILL TYPE THE PROGRAM NAME.
(IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO THE FOLLOWING:

'MAP OF DV11 STATUS'
1500 175000
1502 000300
1504 000226
1506 000062
1510 000226
1512 000062
1514 000226
1516 000062
1520 000226
1522 000062

THE ABOVE IS ONLY AN EXAMPLE! THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADD. 1500 IN THE PROGRAM. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. FOR INFORMATION OF STATUS TABLE SEE SECTION 8.4 FOR HELP.

THE PROGRAM WILL TYPE 'R' AND PROCEED TO RUN THE DIAGNOSTIC

## 4.1 CONTROL SWITCH SETTINGS

NOTE: IF THERE IS NO READ SWR(177570); SWR MAY BE MODIFIED AT LOC 176 OR BY HITTING CONTROL ''G'' <^G> ON CONSOLE TERMINAL.

SW 15 SET: HALT ON ERROR
SW 14 SET: LOOP ON CURRENT TEST
SW 13 SET: INHIBIT ERROR PRINT OUT
SW 12 SET: INHIBIT **ALL** TYPE OUT/BELL ON ERROR.
SW 11 SET: INHIBIT ITERATIONS. (QUICK PASS)
SW 10 SET: ESCAPE TO NEXT TEST
SW 09 SET: LOOP WITH CURRENT DATA
SW 08 SET: CATCH ERROR AND LOOP ON IT
SW 07 SET: USE PREVIOUS STATUS TABLE. CLR-DO AUTO SIZE.
SW 06 SET: RESERVED
SW 05 SET: RESERVED
SW 04 SET: RESERVED
SW 03 SET: RESERVED
SW 02 SET: LOCK ON SELECTED TEST
SW 01 SET: RESTART PROGRAM AT SELECTED TEST
SW 00 SET: RESELECT DV11'S DESIRED ACTIVE.

```
195
196
197                          4.1.2   SWITCH REGISTER RESTRICTIONS
198
199                                  SW 00   RESELECT DV11'S DESIRED ACTIVE.  PLEASE NOTE THAT A  MESSAGE  IS
200                                          TYPED  OUT  FOR  SETTING  THE  SWITCH  REGISTER  EQUAL TO DV11'S
201                                          ACTIVE.   THIS  MEANS  IF  THE  SYSTEM  HAS  FOUR  DV11S;   BITS
202                                          00,01,02,03  WILL  BE  SET  IN  LOC  'DVACTV'  FROM  THE  SWITCH
203                                          REGISTER.     USING     THIS     SWITCH(SW00)     ALTERS     THAT
204                                          LOCATION;THEREFORE  IF FOUR DV11S ARE IN THE SYSTEM ***DO NOT***
205                                          SET SWITCHS GREATER THAN SW 03 IN THE UP POSITION.   THIS  WOULD
206                                          BE A FATAL ERROR.  DO NOT SELECT MORE ACTIVE DV11S THAN HAS BEEN
207                                          GIVEN INFORMATION ABOUT IN TRIAL PROGRAM.
208                          METHOD: A:      LOAD ADDRESS 200
209                                  B:      START WITH SW 00=1
210                                  C:      PROGRAM WILL TYPE MESSAGE
211                                  D:      SET THE BINARY NUMBER OF  DV11S  DESIRED  ACTIVE  EXAMPLE:   1=1
212                                          DV11;   3=2  DV11;   7=3  DV11;  17=4 DV11 37=5 DV11 ETC.  PRESS
213                                          CONTINUE.
214                                  E:      NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/05)
215                                  F:      SET WITH ANY OTHER SWITCH SETTINGS DESIRED.  PRESS CONTINUE.
216
217
218                                  SW 01   RESTART PROGRAM AT SELECTED TEST:       IT IS STRONGLY SUGGESTED
219                                          THAT AT LEAST ONE PASS HAS BEEN MADE BEFORE TRYING TO
220                                          TEST THAT IS NOT IN THE ORDER OF SEQUENCE THE  REASON  BEING  IS
221                                          THAT THE PROGRAM HAS TO CLEAR AREAS AND SET UP PARAMETERS.  ALSO
222                                          WHEN A TEST IS SELECTED ALWAYS START AT THE  VERY  BEGINNING  OF
223                                          THAT TEST.
224
225                                  SW 09   LOOP ON CURRENT DATA:   THIS  SWITCH  WILL  ONLY  WORK  IF  CALL
226                                          'SCOP1'  IS IN THAT TEST.  THE REASON BEING THAT MOST TESTS DEAL
227                                          WITH BLOCKS OF DIFFERENT DATA TO BE SENT OR RECEIVED ALL AT ONCE
228                                          THUS IN BLOCK DATA;  ONE PATTERN CANN'T BE SINGLED OUT.
229
```

```
230
231
232                         4.1.3   SWITCH REGISTER PRIORITYS
233
234                                 ERROR SWITCHES
235
236                                 1.      SW 12   DELETE PRINT OUT/BELL ON ERROR.
237                                 2.      SW 13   DELETE ERROR PRINTOUT.
238                                 3.      SW 15   HALT ON THE ERROR.
239                                 4.      SW 08   GOTO BEGINNING OF THE TEST(ON ERROR).
240                                 5.      SW 10   GOTO NEXT TEST(ON ERROR).
241
242                                 SCOPE SWITCHES
243
244                                 1.      SW 09 (IF ENABLED BY 'SCOP1') ON AN ERROR;  IF AN '*' IS PRINTED
245                                         IN  FRONT  OF  THE  TEST  NO.   (EX.   *TEST  NO.   10 ) SW09 IS
246                                         INCORPORATED IN THAT TEST AND THEREFORE SW09  IS  *USUALLY*  THE
247                                         BEST SWITCH FOR THE SCOPE LOOP (SW14=0, SW10=0, SW09=1, SW08=0).
248                                         IF  SW09  IS  NOT  ENABELED;   AND  THERE  IS  A  *HARD*   ERROR
249                                         (CONSTANT);  SW08 IS BEST.
250                                             (SW14=1,0,  SW10=0,  SW09=0,  SW08=1).   FOR   INTERMITTEMT
251                                         ERRORS;   SW14=1  WILL  LOOP ON TEST REGUARDLESS OF ERROR OR NOT
252                                         ERROR.
253                                             (SW14=1, SW10=0, SW09=0, SW08=1,0)
254                                 2.      SW 14
255                                 3.      SW 11
256
257                         4.2     STARTING ADDRESS
258
259                                 STARTING ADDRESS IS AT 000200 THERE ARE NO OTHER STARTING ADDRESSES  FOR
260                                 THE DV11 DIAGNOSTICS PREVIOUSLY MENTIONED EXCEPT FOR CZDVE WHICH IS:
261                                 000200 FOR THE MODEM CONTROL AND CABLE TESTS AND 000210 FOR  THE  MANUAL
262                                 PARAMETER INPUT PROGRAM.
263
264                                 NOTE:   IF ADDRESS 000042 IS NON-ZERO THE PROGRAM ASSUMES  IT  IS  UNDER
265                                         ACT11  OR  XXDP  CONTROL  AND  WILL  ACT ACCORDINGLY AFTER *ALL*
266                                         AVAILABLE DV11'S ARE TESTED THE PROGRAM WILL RETURN TO 'XXDP' OR
267                                         'ACT-11'.
268
269                         5.      OPERATING PROCEDURE
270
271                                 WHEN PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION  FOUR
272                                 WILL BE PRINTED, AND PROGRAM WILL BEGIN RUNNING THE DIAGNOSTIC.
273
274
```

```
275
276
277                         5.2     PROGRAM AND/OR OPERATOR ACTION
278
279                                 THE TYPICAL APPROACH SHOULD BE
280
281                                 1.      HALT ON ERROR (VIA SW 15=1) WHEN EVER AN ERROR OCCURS.
282                                 2.      CLEAR SW 15.
283                                 3.      SET SW 14:  (LOOP ON THIS TEST)
284                                 4.      SET SW 13:  (INHIBIT ERROR PRINT OUT)
285
286                                 THE TEST NUMBER AND PC WILL BE TYPED OUT AND POSSIBILY AN ERROR MESSAGE
287                                 (THIS DEPENDS ON THE TEST) TO GIVE THE OPERATOR AN IDEA AS TO THE SOURCE
288                                 OF THE PROBLEM.  IF IT IS NECESSARY TO KNOW MORE INFORMATION CONCERNING
289                                 THE ERROR REPORT;  LOOK IN THE LISTING FOR THAT TEST NUMBER WHICH WAS
290                                 TYPED OUT AND THEN NOTE THE PC OF THE ERROR REPORT THIS WAY THE EXACT
291                                 FUNCTIONING OF THE TEST CAN BE INTERPRETED.
292
293                         6.      ERRORS
294
295                                 AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE A TEST NUMBER AND PC TYPED
296                                 OUT AT THE TIME OF AN ERROR (PROVIDING SW 13=0 AND SW 12=0).  IN MOST
297                                 CASES ADDITIONAL INFORMATION WILL BE SUPPLIED TO THE THE ERROR MESSAGE
298                                 WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE ERROR.
299
300                         6.2     ERROR RECOVERY
301
302                                 IF FOR SOME REASON THE DV11 SHOULD 'HANG THE BUS' (GAIN CONTROL OF BUS
303                                 SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT OR POWER DOWN/UP
304                                 IS NECESSARY FOR OPERATOR TO REGAIN CONTROL OF CPU.  IF THIS SHOULD
305                                 HAPPEN;  LOOK IN LOCATION 'TSTNO' (ADDRESS 1224)FOR THE NUMBER OF THE
306                                 TEST THAT WAS RUNNING AT THE TIME OF THE CATASTROPHIC ERROR.  IN THIS
307                                 WAY THE OPERATOR WILL HAVE AN IDEA AS TO WHAT THE DV11 WAS DOING AT THE
308                                 TIME OF THE ERROR.
309
310                         7.      RESTRICTIONS
311
312                         7.1     STARTING RESTRICTIONS
313
314                                 SEE SECTION 4.  (PLEASE)
315                                 STATUS TABLE SHOULD BE VERIFIED REGUARDLESS OF HOW PROGRAM WAS STARTED.
316                                 ALSO IT IS IMPORTANT TO USE THIS LISTING ALONG WITH THE INFORMATION
317                                 PRINTED ON THE TTY TO COMPLETELY ISOLATE PROBLEMS.
318
```

```
319
320
321                         7.2     OPERATING RESTRICTIONS
322
323                                 DV11 TRIAL PROGRAM MUST BE RUN PRIOR TO THE FIRST  AND  ONLY  THE  FIRST
324                                 RUNNING OF ANY DV11 DIAGNOSTIC IF ''AUTO SIZING'' IS NOT USED.
325                                 NOTE:   IF NO PROGRAM OTHER THAN A DV11 DIAGNOSTIC WAS LOADED AFTER DV11
326                                 TRIAL  OR  IF  CORE  MEMORY  HAS NOT BEEN CHANGED;OR IF THERE IS NO DV11
327                                 CONFIGURATION CHANGES;  THE DV11 TRIAL PROGRAM NEED NEVER BE RUN  AGAIN.
328                                 HOWEVER  IF  ANY  OF THE ABOVE HAVE BEEN VIOLATED THE DV11 TRIAL PROGRAM
329                                 MUST BE RUN AGAIN BEFORE RUNNING THE DIAGNOSTICS NOTE:  AN  ALTERNATIVE
330                                 TO  THE  ABOVE IS ATTEMPTING THE 'AUTO SIZING' WHEN PROGRAM IS INITIALLY
331                                 STARTED WITH SW07=0.
332
333                         7.3     HARDWARE CONFIGURATION RESTRICTIONS (SYNC LINE CARDS ONLY)
334
335                                 1.  HARDWARE MUST BE SET TO FULL DUPLEX
336                                 2.  ALL LINES OF A PARTICULAR LINE CARD MUST BE CONFIGURED THE SAME.
337
338                         8.      MISCELLANEOUS
339
340                         8.1     EXECUTION TIME
341
342                                 ALL DV11 DEVICE DIAGNOSTICS WILL GIVE AN 'END PASS'  MESSAGE  (PROVIDING
343                                 NO  ERRORS  AND  SW12=0) WITHIN 4 MINS.  THIS IS ASSUMING SW11=1 (DELETE
344                                 ITERATIONS) IS SET TO GIVE THE FASTEST POSSIBLE EXECUTION.   THE  ACTUAL
345                                 EXECUTION TIME DEPENDS GREATLY ON THE PDP11 CPU CONFIGURATION.
346
347                         8.2     PASS COMPLETE
348
349                                 NOTE:  *EVERY* TIME THE PROGRAM IS STARTED;  THE TESTS WILL  RUN  AS  IF
350                                 SW11  (DELETE  ITERATIONS)  WAS  UP  (=1).  THIS IS TO 'VERIFY NO *HARD*
351                                 ERRORS' AS SOON AS  POSSIBLE.   THEREFORE  THE  FIRST  PASS  -EACH  TIME
352                                 PROGRAM  IS  STARTED- WILL BE A 'QUICK PASS' UNTILL ALL DV11'S IN SYSTEM
353                                 ARE TESTED.  WHEN THE DIAGNOSTIC HAS COMPLETED A PASS THE  FOLLOWING  IS
354                                 AN EXAMPLE OF THE PRINT OUT TO BE EXPECTED.
355
356                                 END PASS CZDVA_  CSR:  175000  VEC:  300  PASSES:  000001
357
358                                 NOTE:    THE NUMBERS FOR CSR AND VEC ARE NOT NECESSARILY THE  VALUES  FOR
359                                          THE DEVICE.  THEY ARE ONLY FOR THIS EXAMPLE.
360                                 NOTE:  CZDVE (MODEM AND CABLE TEST) END PASS MESSAGE IS A LARGE 'END''
361                                          TYPED  OUT  ON  TTY.  PLEASE NOTE THAT EACH CHARACTER PRINTED IS
362                                          ACTUALLY AND ''END PASS'' INDICATION.  THIS WAS USED IN  PLACE  OF
363                                          ''BELL''  BECAUSE  IF  SW12=1 AND AN ERROR OCCURED THE BELL MAY BE
364                                          MISTAKEN FOR END PASS.  THE PASS EXECUTION IS SO FAST  THAT  THE
365                                          STANDARD  END  PASS WAS TOO LENGHTLY.  THEREFORE EACH CHAR IS AN
366                                          ''END PASS AND THE ENTIRE ''END'' IS NOT REQUIRED FOR ACCEPTANCE.
```

```
367
368
369                          8.4     KEY LOCATIONS
370
371                                  RETURN (1212)   CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN
372                                                  ITERATION COUNT IS REACHED OR IF LOOP ON TEST IS
373                                                  ASSERTED.
374                                  NEXT   (1214)   CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PEFORMED.
375                                  TSTNO  (1224)   CONTAINS THE NUMBER OF THE TEST NOW BEING PEFORMED.
376                                  RUN    (1302)   THE BIT IN 'RUN' ALWAYS POINTS ONE PAST THE DV11
377                                                  CURRENTLY BEING TESTED. EXAMPLE: (RUN)
378                                                  1302/0000000001000000 MEANS THAT DV11 NO.05 IS THE DV11
379                                                  NOW RUNNING.
380                                  DVCR00-DVCR17
381                                  DVST00-DVST17
382                                  (1500)-(1736)
383                                                  THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST
384                                                  UP TO 8 (DECIMAL) DV11S SEQUENTIALY. THEY CONTAIN THE
385                                                  CSR,VECTOR AND STATUS CONCERNING THE CONFIGURATION OF
386                                                  EACH DV11.
387                                  DVACTV (1276)   EACH BIT SET IN THIS LOCATION INDICATES THAT THE
388                                                  ASSOCIATED DV11 WILL BE TESTED IN TURN. EXAMPLE:
389                                                  (DVACTV) 1276/0000000000011111 MEANS THAT DV11 NO.
390                                                  00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DVACTV)
391                                                  1276/0000000000010001 MEANS THAT DV11 NO. 00,04 WILL BE
392                                                  TESTED.
393                                  DVSCR  (1356)   CONTAINS THE RECEIVER CSR OF THE CURRENT DV11 UNDER
394                                                  TEST.
395                                  L00.03 (1412)
396                                  L04.07 (1414)
397                                  L08.11 (1416)
398                                  L12.15 (1420)   CONTAINS THE STATUS OF THE CURRENT DV11 UNDER TEST.
399                                          BIT 15  SET:  LINE CARD *NOT INSTALLED (AND WONT BE TESTED)
400                                          BIT 14  SET:  RESERVED
401                                          BIT 13  SET:  RESERVED
402                                          BIT 12  SET:  ONE SYNC, =0: TWO SYNCS.
403                                          BIT 11  SET:  ASYNC LINE CARD, =0 SYNC LINE CARD.
404                                          BIT 10  SET:  RESERVED
405                                          BIT 09   SET: BITS PER CHAR. (USED WITH BIT8)
406                                          BIT 08  SET:  BITS PER CHAR. (USED WITH BIT9)
407                                                BIT09 BIT08 BITS PER CHAR.
408                                                  0     0      8
409                                                  0     1      7
410                                                  1     0      6
411                                                  1     1      5
412                                          BIT 07-00       SYNC "A" FOR SPECIFIED LINE CARD.
413                                                          BITS 07-00 MUST BE ALL ZEROS FOR TESTING OF AN
414                                                          ASYNC LINE CARD.
415
```

```
                        8.4A    MORE ON THAT 'STATUS TABLE' (1500-1736)

                                        'MAP OF DV11 STATUS'
                                        1500    175000
                                        1502    000300
                                        1504    000226
                                        1506    000062
                                        1510    000226
                                        1512    000062
                                        1514    004000
                                        1516    000000
                                        1520    004000
                                        1522    000000


                                SYNC 'A' AND SYNC'B' MUST BE SET TO ZEROS FOR AN ASYNC LINE CARD.
                                THE ABOVE INFORMATION WILL BE REPEATED FOR EACH OF UP TO 8 DV11'S IN THE
                                SYSTEM(THESE WILL FOLLOW UNDER THIS TABLE).  EXPLANATION:
                                1500    175000  THIS IS THE SYSTEM CONTROL REGISTER FOR THE 1ST DV11  IN
                                                THE SYSTEM.
                                1502    000300  THIS IS VECTOR 'A' FOR THE FIRST DV11 IN THE SYSTEM.
                                1504    000226  THIS REPRESENTS 'SYNC A' AND THE SOFTWARE STATUS FOR THE
                                                1ST LINE CARD IN THE 1ST DV11.  THE BITS ARE AS FOLLOWS:

                                        BIT 15  SET:    LINE CARD *NOT INSTALLED (AND WONT BE TESTED)
                                        BIT 14  SET:    RESERVED
                                        BIT 13  SET:    RESERVED
                                        BIT 12  SET:    ONE SYNC, =0:  TWO SYNCS.
                                        BIT 11  SET:    ASYNC LINE CARD, =0 SYNC LINE CARD.
                                        BIT 10  SET:    RESERVED
                                        BIT 09   SET:   BITS PER CHAR.  (USED WITH BIT8)
                                        BIT 08  SET:    BITS PER CHAR.  (USED WITH BIT9)
                                                BIT09 BIT08 BITS PER CHAR.
                                                  0     0       8
                                                  0     1       7
                                                  1     0       6
                                                  1     1       5
                                        BIT 07-00       SYNC 'A' FOR SPECIFIED LINE CARD.
                                1506    000062  THIS REPRESENTS 'SYNC B' FOR THE 1ST LINE CARD.
                                1510    000226  THIS IS 'SYNC A' AND LINE STATUS FOR THE 2ND LINE  CARD.
                                                (FOR BITS DEFINATION SEE EXPLANATION FOR LINE CARD 1).
                                1512    000062  THIS IS 'SYNC B' FOR THE SECOND LINE CARD.
                                1514    000226  THIS IS 'SYNC A' AND LINE STATUS FOR THE 3RD LINE  CARD.
                                                (FOR BITS DEFINATION SEE EXPLANATION FOR LINE CARD 1).
                                1516    000062  THIS IS 'SYNC B' FOR LINE CARD NO.  3.
                                1520    000226  THIS IS 'SYNC A' AND LINE STATUS FOR THE 4TH LINE  CARD.
                                                (FOR BITS DEFINATION SEE EXPLANATION FOR LINE CARD 1).
                                1522    000062  THIS IS SYNC B FOR THE 4TH LINE CARD.

                                THE ABOVE IS REPEATED FOR EACH DV11 IN THE SYSTEM.  THE TABLE IS
                                FILLED  BY  AUTO SIZING OR BY THE MANUAL PARAMETER INPUT PROGRAM
                                AS  DESCRIBED  PREVIOUSLY.   ALSO  IF  DESIRED  BY  USER;    THE
                                LOCATIONS  MAY  BE  ALTERED  BY  HAND  (TOGGLED  IN) TO SUIT THE
                                SPECIFIC CONFIGURATION.
```

```
8.5     *** METHOD OF AUTO SIZING ***

8.5.1   FINDING THE CONTROL STATUS REGISTER.

        THE PROGRAM WILL START AT ADDRESS 175000 AND START 'REFERENCING'
        ADDRESS. IF A NON-EX MEMORY TRAP OCCURS; THE POINTER (HOLDING 175000)
        IS UPDATED BY 10 AND THE ABOVE IS REPEATED UNTIL ADDRESS 175400 IS
        REACHED.  IF  A  'SLAVE  SYNC  RESPONSE' WAS ISSUED BY THE DV11 (OR ANY
        OTHER DEVICE) (NO  NXM  TRAP)(AND  IT(SEL0)WAS=0)  ;  POINTER  PLUS  12
        (SEL12) IS TESTED TO CONTAIN 177777 (MUST BE EXACTLY 177777);  IF A TRAP
        IS ENCOUNTERED OR IF SEL12 DOES NOT CONTAIN 177777 THE ABOVE UPDATING IS
        PERFORMED.  IF SEL12 WAS EQUAL TO 177777 THE POINTER IS STORED AWAY AND
        THE ROUTINE CONTINUES AS ABOVE:
        NOTE:  IF THE PROGRAM DOES NOT FIND YOUR DV11;  SOMETHING IS WRONG  AND
        AUTO SIZING SHOULD NOT BE DONE.

8.5.2   FINDING THE VECTOR

        THE VECTOR AREA (ADDRESS 300-776) IS FILLED WITH THE INSTRUCTION IOT AND
        '.+2' (NEXT ADDRESS).  BIT7 AND BIT6 (RX INTERRUPT AND RX INTERRUPT IE)
        ARE SET INTO DVSCR REGISTER; A DELAY IS MADE AND IF NO INTERRUPT OCCURS
        (BECAUSE  OF  A BAD DV11) THE PROGRAM ASSUMES VECTOR ADDRESS 300 AND THE
        PROBLEM SHOULD BE FIXED IN THE DIAGNOSTIC.  ONCE THE PROBLEM  IS  FIXED;
        THE PROGRAM SHOULD BE SETUP AGAIN TO GET CORRECT VECTOR. IF AN
        INTERRUPT OCCURED; THE ADDRESS TO WHICH THE DV11 INTERRUPTED TO IS PICKED
        UP AND REPORTED AS THE VECTOR.  NOTE:  IF THE VECTOR REPORTED IS NOT THE
        VECTOR SET UP BY YOU;  THERE IS A PROBLEM AND AUTO SIZING SHOULD NOT  BE
        DONE.

8.5.3   PARAMETER ASSUMPTIONS.

        SINCE TOO MUCH HARDWARE WOULD NEED TO BE TURNED ON TO SIZE THE  REST  OF
        THE  PARAMETERS;  THE PROGRAM MUST ASSUME THE REMAINING VARIATIONS.  THE
        RESULT IF NOT TO YOUR SPECIFIC CONFIGURATION  MAY  BE  ALTERED  BY  HANG
        (TOGGLE IN) IS DESIRED.  IN THIS WAY 95% OF THE PARAMETER SETUP WAS DONE
        BY THE PROGRAM AND 5% BY YOU.
        THEREFORE:
        1)      ALL LINE CARDS(4) ARE ASSUMED TO BE INSTALLED.
                SET BIT15 OF STATUS MAP OF ANY (APPROIATE) LINE CARDS MISSING
        2)      TWO SYNCS.
                SET BIT12 IF YOU HAVE A 4 LINE GROUP SET FOR 1 SYNC.
        3)      ADJUST BITS 8 AND 9 IN STATUS MAP FOR YOUR CORRECT CONFIG.
        4)      SYNCHRONOUS LINE CARDS INSTALLED.
                SET BIT11 FOR ASYNC LINE AND ZERO SYNC CHARS.
        5)      SYNC ''A''=226 AND SYNC ''B''=062

    IN ALL ADJUSTMENTS PLEASE REFER TO SECTION 8.4A FOR GREATER DETAIL.
```

```
                                9.0     CHANGE HISTORY

                                        NOTE: HISTORY STARTS WITH REV. C

                                        CZDVAC0 - 1) ADDED TEST 130 TO VERIFY EXTENDED ADDRESS
                                                  BITS 16 AND 17 UTILIZING ALU FUNCTION.

                                                2) MODIFIED FOR COMPATIBILITY WITH COMMON DV11 DIAGNOSTIC
                                                OPERATOR FILE (CZDVXX.OPR), AND SYSMAC FILE (CZDVXX.SML).

                                $
```

```
538
539
540                             ;*CZDVAC-0/<377>/CZDVAC0 BSC TST & ROM INST EX
541                             ;*COPYRIGHT 1972,1979, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
542                             ;*-----------------------------------------------------------------------
543
544                                     ;STARTING PROCEDURE
545                                     ;LOAD PROGRAM
546                                     ;LOAD ADDRESS 000200
547                                     ;PRESS START
548                                     ;PROGRAM WILL TYPE ''CZDVAC-0/<377>/CZDVAC0 BSC TST & ROM INST EX''
549                                     ;PROGRAM WILL TYPE 'R'' TO INDICATE THAT TESTING HAS STARTED
550                                     ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
551                                     ;AND THEN RESUME TESTING
552
553
554                             ;SWITCH REGISTER OPTIONS
555                             ;----------------------
556
557         100000              SW15=100000             ;=1,HALT ON ERROR
558         040000              SW14=40000              ;=1,LOOP ON CURRENT TEST
559         020000              SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
560         010000              SW12=10000              ;=1,DELETE TYPEOUT/BELL ON ERROR.
561         004000              SW11=4000               ;=1,INHIBIT ITERATIONS
562         002000              SW10=2000               ;=1,ESCAPE TO NEXT TEST ON ERROR
563         001000              SW09=1000               ;=1,LOOP WITH CURRENT DATA
564         000400              SW08=400                ;=1,LOOP ON ERROR
565         000200              SW07=200                ;=1, DO ''AUTO SIZING'' ON INITAL START UP.
566         000100              SW06=100
567         000040              SW05=40
568         000020              SW04=20
569         000010              SW03=10
570         000004              SW02=4                  ;LOCK ON TEST SELECT
571         000002              SW01=2                  ;RESTART PROGRAM AT SELECTED TEST
572         000001              SW00=1                  ;RESELECT DV11 DESIRED ACTIVE
573                                                     ;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
```

```
574
575
576                             ;REGISTER DEFINITIONS
577                             ;--------------------
578
579             000000          R0=%0           ;GENERAL REGISTER
580             000001          R1=%1           ;GENERAL REGISTER
581             000002          R2=%2           ;GENERAL REGISTER
582             000003          R3=%3           ;GENERAL REGISTER
583             000004          R4=%4           ;GENERAL REGISTER
584             000005          R5=%5           ;GENERAL REGISTER
585             000006          SP=%6           ;PROCESSOR STACK POINTER
586             000007          PC=%7           ;PROGRAM COUNTER
587
588                             ;LOCATION EQUIVALENCIES
589                             ;----------------------
590
591             177776          PS=177776       ;PROCESSOR STATUS WORD
592             001200          STACK=1200      ;START OF PROCESSOR STACK
593
594             100000          BIT15=100000
595             040000          BIT14=40000
596             020000          BIT13=20000
597             010000          BIT12=10000
598             004000          BIT11=4000
599             002000          BIT10=2000
600             001000          BIT9=1000
601             000400          BIT8=400
602             000200          BIT7=200
603             000100          BIT6=100
604             000040          BIT5=40
605             000020          BIT4=20
606             000010          BIT3=10
607             000004          BIT2=4
608             000002          BIT1=2
609             000001          BIT0=1
610                             ;---------------------------------
611             010000          ALU=BIT12
612             020000          RAM=BIT13
613             030000          XFR=BIT13+BIT12
614             040000          NPR=BIT14
615             050000          S.C=BIT14+BIT12
616             060000          BCC=BIT14+BIT13
617             070000          BRB=BIT14+BIT13+BIT12
618                             ;---------------------------------
619
620
```

```
                        ;:**********************************************************************
                        ;----------------------------------------------------------------------
                                ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
                                ;THE STANDARD "TRAP CATCHER" IS PLACED
                                ;BETWEEN ADDRESS 0 TO ADDRESS 776.
                                ;IT LOOKS LIKE 'PC+2 HALT'.
                        ;----------------------------------------------------------------------
                        ;:**********************************************************************

                000000  .=0
                                ;STANDARD INTERRUPT VECTORS
                                ;-------------------------

                000024  .=24
000024  004402                  .PFAIL                  ;POWER FAIL HANDLER
000026  000340                  340                     ;SERVICE AT LEVEL 7
000030  004002                  .HLT                    ;ERROR HANDLER
000032  000340                  340                     ;SERVICE AT LEVEL 7
000034  003750                  .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                  340                     ;SERVICE AT LEVEL 7
                000040  .=40
000040  000001                  .BLKW 1                 ;SAVE FOR ACT-11 OR DDP2
000042  000001                  .BLKW 1                 ;RETURN ADDRESS IF UNDER ACT-11 OR DDP2
000044  000001                  .BLKW 1                 ;SAVE FOR ACT-11 OR DDP2
000046  002560                  LOGICAL                 ;FOR USE WITH ACT-11 OR DDP2

                000174  .=174
000174  000000          LIGHT:  0
                000176  .=176
000176  000000          SSWR:   0

                000200  .=200
000200  000137  001742          JMP     .START          ;GO TO START OF PROGRAM


                001000  .=1000
001000  005377  055103  053104  MTITLE: .ASCIZ  <377><12>/CZDVAC-O/<377>/CZDVACO BSC TST & ROM INST EX/<377>
(2)
                001200  .=1200
001200                  LIGHTS:
001200  177570                  177570
001202  177570          SWR:    177570
                                ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
                                ;--------------------------------------------------

001204  177560          TKCSR:  177560                  ;TELETYPE KEYBOARD CONTROL REGISTER
001206  177562          TKDBR:  177562                  ;TELETYPE KEYBOARD DATA BUFFER
001210  177564          TPCSR:  177564                  ;TELEPRINTER CONTROL REGISTER
001212  177566          TPDBR:  177566                  ;TELEPRINTER DATA BUFFER

                                ;PROGRAM CONTROL PARAMETERS
                                ;--------------------------

001214  000000          RETURN: 0                       ;SCOPE ADDRESS FOR LOOP ON TEST
001216  000000          NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
001220  000000          LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA
```

```
                                ICOUNT: 3                       ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
                                LPCNT:  0                       ;NUMBER OF ITEREATIONS COMPLETED
                                TSTNO:  0                       ;NUMBER OF TEST IN PROGRESS
                                PASCNT: 0                       ;NUMBER OF PASSES COMPLETED
                                ERRCNT: 0                       ;TOTAL NUMBER OF ERRORS
                                LSTERR: 0                       ;PC OF LAST ERROR CALL

                                        ;PROGRAM VARIABLES
                                        ;-----------------

                                STAT:   0                       ;DV STATUS WORD STORAGE
                                SYNCX:  0
                                CLKX:   0
                                MASKX:  0
                                TEMP1:  0                       ;TEMPORARY STORAGE
                                TEMP2:  0                       ;TEMPORARY STORAGE
                                TEMP3:  0                       ;TEMPORARY STORAGE
                                TEMP4:  0                       ;TEMPORARY STORAGE
                                TEMP5:  0                       ;TEMPORARY STORAGE
                                SAVR0:  0                       ;R0 STORAGE
                                SAVR1:  0                       ;R1 STORAGE
                                SAVR2:  0                       ;R2 STORAGE
                                SAVR3:  0                       ;R3 STORAGE
                                SAVR4:  0                       ;R4 STORAGE
                                SAVR5:  0                       ;R5 STORAGE
                                SAVSP:  0                       ;STACK POINTER STORAGE
                                SAVPC:  0                       ;PROGRAM COUNTER STORAGE
                                DVACTV: .BLKB 1                 ;DV11'S SELECTED ACTIVE.
                                DVNUM:  .BLKB 1                 ;OCTAL NUMBER OF DV11'S.
                                SAVACT: .BLKB 1                 ;ORIGINAL ACTV.  DEVICES.
                                SAVNUM: .BLKB 1                 ;WORKABLE NUMBER.
                                RUN:    .BLKB 1                 ;POINTER ONE PAST RUNNING DEVICE.
                                .EVEN
                                CREAM:  DV.MAP                  ;TABLE POINTER.
```

```
                        ;PROGRAM CONTROL FLAGS
                        ;---------------------

001310  000     INIFLG: .BYTE   0                ;PROGRAM INITIALIZATION FLAG
001311  000     ERRFLG: .BYTE   0                ;ERROR OCCURED FLAG
001312  000     LOKFLG: .BYTE   0                ;LOCK ON CURRENT TEST FLAG
001313  000     QV.FLG: .BYTE   0                ;QUICK VERIFY FLAG.
                                                 ;ON FIRST PASS OF EACH DV11 ITERATIONS WILL BE SUPPRESSE

                .EVEN
000000          $Y=0

                        ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
                        ;POINTERS TO SUBROUTINES CAN BE FOUND
                        ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS

                ;:*****************************************************************
                ;-----------------------------------------------------------------
001314          .TRPTAB:
104400          SCOPE=TRAP+0            ;CALL TO SCOPE LOOP AND ITERATION HANDLER
001314  002634          .SCOPE
104401          SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
001316  003020          .SCOP1
104402          TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
001320  003044          .TYPE
104403          INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
001322  003120          .INSTR
104404          INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
001324  003224          .INSTER
104405          PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
001326  003244          .PARAM
104406          SAV05=TRAP+6            ;CALL TO REGISTER SAVE ROUTINE
001330  003444          .SAV05
104407          RES05=TRAP+7            ;CALL TO REGISTER RESTORE ROUTINE
001332  003504          .RES05
104410          CONVRT=TRAP+10          ;CALL TO DATA OUTPUT ROUTINE
001334  003536          .CONVRT
104411          CNVRT=TRAP+11           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
001336  003542          .CNVRT
104412          MSTCLR=TRAP+12          ;CALL TO ISUE A MASTER CLEAR
001340  004556          .MSTCLR
104413          RAMCLR=TRAP+13          ;CALL TO CLEAR THE RAMS
001342  004516          .RAMCLR
104414          DELAY=TRAP+14           ;CALL TO VARIABLE DELAY COUNTER
001344  004476          .DELAY
104415          ROMCLK=TRAP+15          ;CALL TO CLOCK ROM ONCE
001346  004566          .ROMCLK
104416          DATACLK=TRAP+16         ;CALL TO CLK DATA
001350  004576          .DATACLK

                ;-----------------------------------------------------------------
                ;:*****************************************************************
```

```
                                        ;DV11 VECTOR AND REGISTER INDIRECT POINTERS

001352  000000          DVRVEC: 0             ;POINTER TO DV11 RECEIVER INTERRUPT VECTOR
001354  000000          DVRLVL: 0             ;POINTER TO DV11 RECEIVER INTERRUPT SERVICE PS
001356  000000          DVTVEC: 0             ;POINTER TO DV11 TRANSMITTER INTERRUPT VECTOR
001360  000000          DVTLVL: 0             ;POINTER TO DV11 TRANSMITTER INTERRUPT SERVICE PS
001362  000000          DVSCR:  0             ;POINTER TO DV11 SYSTEM CONTROL REGISTER
001364  000000          DVSCRH: 0             ;POINTER TO DV11 SYSTEM CONTROL REGISTER HIGH BYTE.
001366  000000          DVRIC:  0             ;POINTER TO DV11 NEXT RECEIVED CHARACTER REGISTER
001370  000000          DVLCR:  0             ;POINTER TO DV11 LINE PRAMETER REGISTER
001372  000000          DVSRS:  0             ;POINTER TO DV11 SECONDARY REGISTER SELECT REGISTER
001374  000000          DVSRSH: 0             ;POINTER TO DV11 SECONDARY REGISTER SELECT HIGH BYTE.
001376  000000          DVSRA:  0             ;POINTER TO DV11 SECONDARY REGISTER ACCESS REGISTER
001400  000000          DVSFR:  0             ;POINTER TO DV11 SPECIAL FUNCTIONS REGISTER
001402  000000          DVNSR:  0             ;POINTER TO DV11 NPR STATUS REGISTER
001404  000000          RESV16: 0             ;POINTER TO RESERVED REGISTER.


                                        ;DV11 CONTROL INDICATORS FOR CURRENT DV11 UNDER TEST
                                        ;-------------------------------------------------

001406  000000          MASK.A: .WORD 000     ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 00-03
001410  000000          MASK.B: .WORD 000     ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 04-07
001412  000000          MASK.C: .WORD 000     ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 08-11
001414  000000          MASK.D: .WORD 000     ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 12-15

001416      010         CLK.A:  .BYTE 8.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 00-03
001417      010         CLK.B:  .BYTE 8.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 04-07
001420      010         CLK.C:  .BYTE 8.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 08-11
001421      010         CLK.D:  .BYTE 8.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 12-15

001422  000000          L00.03: 000000        ;PARAMETERS FOR LINES 00-03
001424  000000          L04.07: 000000        ;PARAMETERS FOR LINES 04-07
001426  000000          L08.11: 000000        ;PARAMETERS FOR LINES 08-11
001430  000000          L12.15: 000000        ;PARAMETERS FOR LINES 12-15

001432  000000          SYNC2A: 000000        ;SYNC 2
001434  000000          SYNC2B: 000000        ;
001436  000000          SYNC2C: 000000        ;
001440  000000          SYNC2D: 000000        ;


                                        ;SUMMARY
                                        ;-------
                        ;       MASK.X          040     5 BITS PER CHAR.
                        ;                       100     6 BITS PER CHAR.
                        ;                       200     7 BITS PER CHAR.
                        ;                       400     8 BITS PER CHAR.

                        ;       CLK.X           005     5 BITS PER CHAR.
                        ;                       006     6 BITS PER CHAR.
                        ;                       007     7 BITS PER CHAR.
                        ;                       010     8 BITS PER CHAR.
                        ;       IF PARITY IS ENABLED; ADD PLUS ONE TO THE ABOVE "CLK.X"
                        ;       FOR EACH GROUP THAT PARITY IS ENABLED.
```

```
                                        ;DV11 STATUS TABLE AND ADDRESS ASSIGNMENTS
                                        ;-----------------------------------------

          001500                        .=1500
001500                                  DV.MAP:
001500    000001                        DVCR00: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 00
001502    000001                        DVTR00: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 00
001504    000001                        DV00.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 00
001506    000001                        SYNA00: .BLKW 1         ;SYNC TWO
001510    000001                        DV00.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 00
001512    000001                        SYNB00: .BLKW 1         ;SYNC TWO
001514    000001                        DV00.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 00
001516    000001                        SYNC00: .BLKW 1         ;SYNC TWO
001520    000001                        DV00.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 00
001522    000001                        SYND00: .BLKW 1         ;SYNC TWO

001524    000001                        DVCR01: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 01
001526    000001                        DVTR01: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 01
001530    000001                        DV01.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 01
001532    000001                        SYNA01: .BLKW 1         ;SYNC TWO
001534    000001                        DV01.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 01
001536    000001                        SYNB01: .BLKW 1         ;SYNC TWO
001540    000001                        DV01.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 01
001542    000001                        SYNC01: .BLKW 1         ;SYNC TWO
001544    000001                        DV01.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 01
001546    000001                        SYND01: .BLKW 1         ;SYNC TWO

001550    000001                        DVCR02: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 02
001552    000001                        DVTR02: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 02
001554    000001                        DV02.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 02
001556    000001                        SYNA02: .BLKW 1         ;SYNC TWO
001560    000001                        DV02.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 02
001562    000001                        SYNB02: .BLKW 1         ;SYNC TWO
001564    000001                        DV02.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 02
001566    000001                        SYNC02: .BLKW 1         ;SYNC TWO
001570    000001                        DV02.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 02
001572    000001                        SYND02: .BLKW 1         ;SYNC TWO

001574    000001                        DVCR03: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 03
001576    000001                        DVTR03: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 03
001600    000001                        DV03.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 03
001602    000001                        SYNA03: .BLKW 1         ;SYNC TWO
001604    000001                        DV03.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 03
001606    000001                        SYNB03: .BLKW 1         ;SYNC TWO
001610    000001                        DV03.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 03
001612    000001                        SYNC03: .BLKW 1         ;SYNC TWO
001614    000001                        DV03.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 03
001616    000001                        SYND03: .BLKW 1         ;SYNC TWO

001620    000001                        DVCR04: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 04
001622    000001                        DVTR04: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 04
001624    000001                        DV04.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 04
001626    000001                        SYNA04: .BLKW 1         ;SYNC TWO
001630    000001                        DV04.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 04
001632    000001                        SYNB04: .BLKW 1         ;SYNC TWO
001634    000001                        DV04.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 04
```

```
001636  000001                  SYNC04: .BLKW 1         ;SYNC TWO
001640  000001                  DV04.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 04
001642  000001                  SYND04: .BLKW 1         ;SYNC TWO

001644  000001                  DVCR05: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 05
001646  000001                  DVTR05: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 05
001650  000001                  DV05.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 05
001652  000001                  SYNA05: .BLKW 1         ;SYNC TWO
001654  000001                  DV05.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 05
001656  000001                  SYNB05: .BLKW 1         ;SYNC TWO
001660  000001                  DV05.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 05
001662  000001                  SYNC05: .BLKW 1         ;SYNC TWO
001664  000001                  DV05.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 05
001666  000001                  SYND05: .BLKW 1         ;SYNC TWO

001670  000001                  DVCR06: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 06
001672  000001                  DVTR06: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 06
001674  000001                  DV06.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 06
001676  000001                  SYNA06: .BLKW 1         ;SYNC TWO
001700  000001                  DV06.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 06
001702  000001                  SYNB06: .BLKW 1         ;SYNC TWO
001704  000001                  DV06.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 06
001706  000001                  SYNC06: .BLKW 1         ;SYNC TWO
001710  000001                  DV06.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 06
001712  000001                  SYND06: .BLKW 1         ;SYNC TWO

001714  000001                  DVCR07: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 07
001716  000001                  DVTR07: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 07
001720  000001                  DV07.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 07
001722  000001                  SYNA07: .BLKW 1         ;SYNC TWO
001724  000001                  DV07.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 07
001726  000001                  SYNB07: .BLKW 1         ;SYNC TWO
001730  000001                  DV07.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 07
001732  000001                  SYNC07: .BLKW 1         ;SYNC TWO
001734  000001                  DV07.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 07
001736  000001                  SYND07: .BLKW 1         ;SYNC TWO

001740  000000                  DV.END: 000000

                                ;PROGRAM INITIALIZATION
                                ;LOCK OUT INTERRUPTS
                                ;SET UP PROCESSOR STACK
                                ;SET UP POWER FAIL VECTOR
                                ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
                                ;TYPE TITLE MESSAGE

001742  012737  000340  177776  .START: MOV     #340,PS             ;LOCK OUT INTERRUPTS
001750  012706  001200          MOV     #STACK,SP           ;SET UP STACK
001754  012737  004402  000024  MOV     #.PFAIL,@#24        ;SET UP POWER FAIL VECTOR
001762  113737  001301  001303  MOVB    DVNUM,SAVNUM        ;SAVE NUMBER OF DEVICES IN SYSTEM.
001770  005037  001230          CLR     PASCNT              ;CLEAR PASS COUNT
001774  105037  001311          CLRB    ERRFLG              ;CLEAR ERROR FLAG
002000  105037  001313          CLRB    QV.FLG              ;ZERO QUICK VERIFY FLAG
002004  012737  001500  001306  MOV     #DV.MAP,CREAM       ;GET MAP POINTER.
002012  112737  000001  001304  MOVB    #1,RUN              ;POINT POINTER TO FIRST DEVICE.
002020  005037  001232          CLR     ERRCNT              ;CLEAR ERROR COUNT
```

```
   928  002024  005037  001234                CLR     LSTERR                  ;CLEAR LAST ERROR POINTER
   929  002030  012737  000001  001226        MOV     #1,TSTNO                ;SET UP FOR TEST 1
   930  002036  012737  001742  001214        MOV     #.START,RETURN          ;SET UP FOR POWER FAIL BEFORE
   931                                                                        ;TESTING STARTS
   932  002044  105737  001310                TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
   933  002050  001063                        BNE     1$                      ;BR IF YES
   934  002052  013746  000004                MOV     4,-(SP)                 ;
   935  002056  013746  000006                MOV     6,-(SP)                 ;
   936  002062  005037  000006                CLR     6                       ;
   937  002066  012737  002104  000004        MOV     #80$,4                  ;
   938  002074  005777  177102                TST     @SWR                    ;
   939  002100  000240                        NOP                             ;
   940  002102  000407                        BR      81$                     ;
   941  002104  022626                80$:    CMP     (SP)+,(SP)+             ;
   942  002106  012737  000174  001200        MOV     #LIGHT,LIGHTS           ;
   943  002114  012737  000176  001202        MOV     #SSWR,SWR               ;
   944  002122  012637  000006        81$:    MOV     (SP)+,6                 ;
   945  002126  012637  000004                MOV     (SP)+,4                 ;
   946  002132  104402  001000                TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
   947  002136  105137  001310                COMB    INIFLG                  ;IF NOT SET FLAG AND DO
   948  002142  105777  177034                TSTB    @SWR                    ;BIT7=1??
   949  002146  100402                        BMI     16$                     ;BR IF NO AUTO SIZE
   950  002150  004737  006626                JSR     PC,CSRMAP               ;GO DO THE AUTO SIZE
   951  002154  104402  005461        16$:    TYPE    ,XHEAD          ;TYPE HEADER
   952  002160  012737  001500  001246        MOV     #DV.MAP,TEMP1           ;SET POINTER
   953  002166  017737  177054  001250  5$:   MOV     @TEMP1,TEMP2            ;SET DATA
   954  002174  022737  177777  001250        CMP     #177777,TEMP2           ;ALL DONE?
   955  002202  001406                        BEQ     1$                      ;BR IF YES
   956  002204  104410                        CONVRT                          ;
   957  002206  005506                        XSTATQ                          ;
   958  002210  062737  000002  001246        ADD     #2,TEMP1                ;UPDATE POINTER
   959  002216  000763                        BR      5$
   960  002220  005737  000042        1$:     TST     @#42                    ;IS PROGRAM RUNNING UNDER MONITOR
   961  002224  001030                        BNE     3$                      ;BR IF YES
   962  002226  032777  000001  176746        BIT     #SW00,@SWR              ;SELECT SPECIFIC DEVICES??
   963  002234  001424                        BEQ     3$                      ;BR IF NO.
   964  002236  104402  005402                TYPE    ,MNEW                   ;TYPE THE MESSAGE.
   965  002242  005000                        CLR     R0                      ;ZERO DATA LIGHTS
   966  002244  000000                        HALT                            ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
   967  002246  127737  176730  001302        CMPB    @SWR,SAVACT             ;IS THE NUMBER VALID?
   968  002254  101404                        BLOS    2$                      ;BR IF NUMBER IS OK.
   969  002256  104402  005243                TYPE    ,MERR3                  ;TELL USER OF INVALID NUMBER.
   970  002262  000000                        HALT                            ;STOP EVERY THING.
   971  002264  000776                        BR      .-2                     ;RESTART THE PROGRAM AGAIN.
   972  002266  117737  176710  001300  2$:    MOVB    @SWR,DVACTV             ;GET NEW DEVICE PATTERN
   973  002274  113700  001300                MOVB    DVACTV,R0               ;SHOW THE USER WHAT HE SELECTED.
   974  002300  042700  177400                BIC     #^C<377>,R0             ;USE ONLY LOW BYTE.
   975  002304  000000                        HALT                            ;CONTINUE DYNAMIC SWITCHES.
   976  002306  012700  000300        3$:     MOV     #300,R0                 ;PREPARE TO CLEAR THE FLOATING
   977  002312  012701  000302                MOV     #302,R1                 ;VECTOR AREA. 300-776
   978  002316  010120                4$:     MOV     R1,(R0)+                ;START PUTTING 'PC+2 - HALT''
   979  002320  005021                        CLR     (R1)+                   ;IN VECTOR AREA.
   980  002322  022021                        CMP     (R0)+,(R1)+             ;POP POINTERS
   981  002324  022700  001000                CMP     #1000,R0                ;ALL DONE??
   982  002330  001372                        BNE     4$                      ;BR IF NO.
   983
```

```
                                        ;TEST START AND RESTART
                                        ;----------------------

 002332  012737  000340  177776  .BEGIN: MOV     #340,PS                 ;LOCK OUT INTERRUPTS
 002340  012706  001200                  MOV     #STACK,SP               ;SET UP STACK
 002344  005737  000042                  TST     @#42                    ;IS PROGRAM UNDER MONITOR CONTROL
 002350  001023                          BNE     3$                      ;BR IF YES
 002352  032777  000004  176622          BIT     #BIT2,@SWR              ;CHECK FOR LOCK ON TEST
 002360  001411                          BEQ     1$                      ;BR IF NO LOCK DESIRED.
 002362  104402  005301                  TYPE    ,MLOCK                  ;TYPE LOCK SELECTED.
 002366  012737  000240  002702          MOV     #NOP,TTST               ;ADJUST SCOPE ROUTINE.
 002374  012737  000240  002704          MOV     #NOP,TTST+2             ;SET UP TO LOCK
 002402  000406                          BR      2$                      ;CONTINUE ALONG.
 002404  013737  003014  002702  1$:     MOV     BRW,TTST                ;PREPARE NORMAL SCOPE ROUTINE
 002412  013737  003016  002704          MOV     BRX,TTST+2              ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
 002420                          2$:
 002420  012737  005666  001214  3$:     MOV     #CYCLE,RETURN           ;START AT ''CYCLE'' FIND WHICH DEVICE TO TEST
 002426  104402  005171          4$:     TYPE    ,MR                     ;TYPE R
 002432  000177  176556                  JMP     @RETURN                 ;START TESTING
```

```
                                        ;END OF PASS
                                        ;TYPE NAME OF TEST
                                        ;UPDATE PASS COUNT
                                        ;CHECK FOR EXIT TO ACT-11
                                        ;RESTART TEST

002436  000005                  .EOP:   RESET                   ;MAKE THE  WORLD CLEAN AGAIN.
002440  005037  001234                  CLR     LSTERR          ;CLEAR LAST ERROR PC
002444  105037  001311                  CLRB    ERRFLG          ;CLEAR ERROR FLAG
002450  005237  001230                  INC     PASCNT          ;UPDATE PASS COUNT
002454  013777  001230  176516          MOV     PASCNT,@LIGHTS  ;DISPLAY PASS COUNT
002462  104402  005145                  TYPE    ,MEPASS         ;TYPE END PASS
002466  104402  005330                  TYPE    ,MCSRX          ;TYPE CSR
002472  104411  002604                  CNVRT   ,XCSR           ;SHOW IT
002476  104402  005336                  TYPE    ,MVECX          ;TYPE VECTOR
002502  104411  002612                  CNVRT   ,XVEC           ;SHOW IT
002506  104402  005344                  TYPE    ,MPASSX         ;TYPE PASSES
002512  104411  002620                  CNVRT   ,XPASS          ;SHOW IT
002516  104402  005355                  TYPE    ,MERRX          ;TYPE ERRORS
002522  104411  002626                  CNVRT   ,XERR           ;SHOW IT
002526  105337  001303                  DECB    SAVNUM          ;ARE ALL DEVICES TESTED?
002532  001017                          BNE     RESTRT          ;BR IF NO.
002534  112737  000377  001313          MOVB    #377,QV.FLG     ;SET THE QUICK VERIFY FLAG.
002542  113737  001301  001303          MOVB    DVNUM,SAVNUM    ;RESTORE THE COUNT
002550  013701  000042                  MOV     @#42,R1         ;CHECK FOR ACT-11 OR DDP
002554  001406                          BEQ     RESTRT          ;IF NOT, CONTINUE TESTING
002556  000005                          RESET                   ;STOP THE SHOW--CLEAR THE WORLD
002560                          LOGICAL:
002560  004711                          JSR     PC,(R1)
002562  000240                          NOP
002564  000240                          NOP
002566  000240                          NOP
002570  000240                          NOP
002572  012737  005666  001214  RESTRT: MOV     #CYCLE,RETURN
002600  000137  005666                  JMP     CYCLE
002604  000001                  XCSR:   1
002606     006     002                  .BYTE   6,2
002610  001362                          DVSCR
002612  000001                  XVEC:   1
002614     003     002                  .BYTE   3,2
002616  001352                          DVRVEC
002620  000001                  XPASS:  1
002622     006     002                  .BYTE   6,2
002624  001230                          PASCNT
002626  000001                  XERR:   1
002630     006     002                  .BYTE   6,2
002632  001232                          ERRCNT

                                        ;SCOPE LOOP AND INTERATION HANDLER
                                        ;--------------------------------

002634                          .SCOPE:
002634  022737  177570  001202          CMP     #177570,SWR     ;IS THERE A REAL SWR?
002642  001411                          BEQ     64$             ;BR IF YES
002644  017746  176336                  MOV     @TKDBR,-(SP)    ;SAVE KEYBOARD CHAR
002650  042716  000200                  BIC     #BIT7,(SP)      ;CLEAR PARITY BIT
```

```
 1059  002654  122726  000007              CMPB    #7,(SP)+        ;WAS IT CNTRL 'G' ?
 1060  002660  001002                      BNE     .+6             ;BR IF NO.
 1061  002662  004737  004640              JSR     PC,SERV.G       ;SERVICE ''CNTRL 'G'''.
 1062  002666  005037  001234      64$:    CLR     LSTERR          ;CLEAR LAST ERROR PC.
 1063  002672  010016                      MOV     R0,(SP)         ;SAVE R0 ON THE STACK
 1064  002674  032777  040000  176300      BIT     #BIT14,@SWR     ;'LOOP ON THIS TEST'?
 1065  002702  001407              TTST:   BEQ     1$              ;BR IF NO.  (IF LOCK SW01=1; THIS LOC =240)
 1066  002704  000437                      BR      3$              ;GOTO 3$    (IF LOCK SW01=1; THIS LOC =240)
 1067  002706  105777  176272              TSTB    @TKCSR          ;KEYBOAD DONE?
 1068  002712  100034                      BPL     3$              ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
 1069  002714  017700  176266              MOV     @TKDBR,R0       ;CLEAR DONE BIT
 1070  002720  000415                      BR      2$              ;CONTINUE
 1071  002722  032777  004000  176252  1$: BIT     #SW11,@SWR      ;DELETE ITERATION?  (QUICK PASS)
 1072  002730  001011                      BNE     2$              ;BR IF YES
 1073  002732  105737  001313              TSTB    QV.FLG          ;HAVE PASSES BEECOMPLETED?
 1074  002736  001406                      BEQ     2$              ;BR IF QUICK PASS.
 1075  002740  005237  001224              INC     LPCNT           ;UPDATE ITERATION COUNTER
 1076  002744  023737  001224  001222      CMP     LPCNT,ICOUNT    ;ARE ALL ITERATIONS DONE??
 1077  002752  001014                      BNE     3$              ;BR IF NOT YET
 1078  002754  105037  001311      2$:     CLRB    ERRFLG          ;PREPARE FOR NEW TEST
 1079  002760  005037  001224              CLR     LPCNT           ;START ICOUNTER AT 0
 1080  002764  005037  001220              CLR     LOCK
 1081  002770  012737  000020  001222      MOV     #20,ICOUNT      ;RESET ITERATIONS
 1082  002776  013737  001216  001214      MOV     NEXT,RETURN     ;GET NEXT TEST
 1083  003004  011600              3$:     MOV     (SP),R0         ;POP R0 OFF OF THE STACK
 1084  003006  022626                      POP2SP                  ;FAKE AN 'RTI''
 1085  003010  000177  176200              JMP     @RETURN         ;GO DO THE TEST
 1086  003014  001407              BRW:    1407
 1087  003016  000437              BRX:    437
 1088
 1089                                      ;CHECK FOR FREEZE ON CURRENT DATA
 1090                                      ;-------------------------------
 1091
 1092  003020  032777  001000  176154  .SCOP1: BIT     #SW09,@SWR      ;IS SW09=1(SET)?
 1093  003026  001405                      BEQ     1$              ;BR IF NOT SET.
 1094  003030  005737  001220              TST     LOCK
 1095  003034  001402                      BEQ     1$
 1096  003036  013716  001220              MOV     LOCK,(SP)       ;GOTO THE ADDRESS IN LOCK.
 1097  003042  000002              1$:     RTI                     ;GO BACK.
 1098
 1099                                      ;TELETYPE OUTPUT ROUTINE
 1100                                      ;----------------------
 1101
 1102  003044  010546              .TYPE:  MOV     R5,-(SP)        ;SAVE R5 ON THE STACK.
 1103  003046  017605  000002              MOV     @2(SP),R5       ;GET ADDRESS OF MESSAGE.
 1104  003052  062766  000002  000002      ADD     #2,2(SP)        ;POP OVER ADDRESS.
 1105  003060  032777  010000  176114  1$: BIT     #SW12,@SWR      ;INHIBIT ALL PRINT OUT??
 1106  003066  001012                      BNE     3$              ;BR IF NO PRINT OUT WANTED (SW12=1)
 1107  003070  105715                      TSTB    (R5)            ;IS NUMBER MINUS? (MSB=1(BIT7))
 1108  003072  100002                      BPL     2$              ;BR IF NUMBER IS PLUS
 1109  003074  104402  005104              TYPE    ,MCRLF          ;TYPE A CR/LF!
 1110  003100  105777  176104      2$:     TSTB    @TPCSR          ;TTY READY?
 1111  003104  100375                      BPL     2$              ;BR IF NO.
 1112  003106  112577  176100              MOVB    (R5)+,@TPDBR    ;PRINT CURRENT CHAR.
 1113  003112  001362                      BNE     1$              ;IF NOT ZERO KEEP PRINTING!
 1114  003114  012605              3$:     MOV     (SP)+,R5        ;END OF OUTPUT. RESTORE R5
```

```
1115  003116  000002                          RTI                     ;GO HOME
1116                                          ;-----------------------
1117
1118  003120  010346                  .INSTR: MOV     R3,-(SP)        ;SAVE R3 ON STACK
1119  003122  010446                          MOV     R4,-(SP)        ;SAVE R4 ON STACK
1120  003124  017637  000004  003142          MOV     @4(SP),,MSG
1121  003132  062766  000002  000004          ADD     #2,4(SP)
1122  003140  104402                  .INST1: TYPE
1123  003142  000000                  .MSG:   0
1124  003144  012704  005520                  MOV     #INBUF,R4
1125  003150  012703  000007                  MOV     #7,R3
1126  003154  105777  176024          1$:     TSTB    @TKCSR
1127  003160  100375                          BPL     1$
1128  003162  117714  176020                  MOVB    @TKDBR,(R4)
1129  003166  142714  000200                  BICB    #200,(R4)
1130  003172  122427  000015                  CMPB    (R4)+,#15
1131  003176  001417                          BEQ     INSTR2
1132  003200  105777  176004          2$:     TSTB    @TPCSR
1133  003204  100375                          BPL     2$
1134  003206  017777  175774  175776          MOV     @TKDBR,@TPDBR
1135  003214  005303                          DEC     R3
1136  003216  001356                          BNE     1$
1137  003220  012604                          MOV     (SP)+,R4
1138  003222  012603                          MOV     (SP)+,R3
1139  003224  104402  005100          .INSTE: TYPE    ,MQM
1140  003230  010346                          MOV     R3,-(SP)
1141  003232  010446                          MOV     R4,-(SP)
1142  003234  000741                          BR      .INST1
1143  003236  012604                  INSTR2: MOV     (SP)+,R4        ;RESTORE R4
1144  003240  012603                          MOV     (SP)+,R3        ;RESTORE R3
1145  003242  000002                          RTI
1146
1147                                          ;CONVERT ASCII STRING TO OCTAL
1148                                          ;-----------------------------
1149
1150  003244  010546                  .PARAM: MOV     R5,-(SP)
1151  003246  010446                          MOV     R4,-(SP)
1152  003250  016605  000004                  MOV     4(SP),R5
1153  003254  012537  003434                  MOV     (R5)+,LOLIM
1154  003260  012537  003436                  MOV     (R5)+,HILIM
1155  003264  012537  003440                  MOV     (R5)+,DEVADR
1156  003270  112537  003442                  MOVB    (R5)+,LOBITS
1157  003274  112537  003443                  MOVB    (R5)+,ADRCNT
1158  003300  010566  000004                  MOV     R5,4(SP)
1159  003304  005005                  PARAM1: CLR     R5
1160  003306  012704  005520                  MOV     #INBUF,R4
1161  003312  122714  000015                  CMPB    #15,(R4)
1162  003316  001420                          BEQ     PARERR
1163  003320  121427  000060          1$:     CMPB    (R4),#60
1164  003324  002415                          BLT     PARERR
1165  003326  121427  000067                  CMPB    (R4),#67
1166  003332  003012                          BGT     PARERR
1167  003334  142714  000060                  BICB    #60,(R4)
1168  003340  152405                          BISB    (R4)+,R5
1169  003342  122714  000015                  CMPB    #15,(R4)
1170  003346  001406                          BEQ     LIMITS
```

```
1171  003350  006305                    ASL     R5
1172  003352  006305                    ASL     R5
1173  003354  006305                    ASL     R5
1174  003356  000760                    BR      1$
1175  003360  104404              PARERR: INSTER
1176  003362  000750                    BR      PARAM1
1177
1178                                    ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
1179                                    ;--------------------------------------
1180
1181  003364  020537  003436      LIMITS: CMP     R5,HILIM
1182  003370  101373                    BHI     PARERR
1183  003372  020537  003434            CMP     R5,LOLIM
1184  003376  103770                    BLO     PARERR
1185  003400  133705  003442            BITB    LOBITS,R5
1186  003404  001365                    BNE     PARERR
1187
1188                                    ;STORE NUMBER AT SPECIFIED ADDRESS
1189
1190  003406  013704  003440            MOV     DEVADR,R4
1191  003412  010524              1$:   MOV     R5,(R4)+
1192  003414  062705  000002            ADD     #2,R5
1193  003420  105337  003443            DECB    ADRCNT
1194  003424  001372                    BNE     1$
1195  003426  012604                    MOV     (SP)+,R4
1196  003430  012605                    MOV     (SP)+,R5
1197  003432  000002                    RTI
1198  003434  000000              LOLIM:  0
1199  003436  000000              HILIM:  0
1200  003440  000000              DEVADR: 0
1201  003442  000000              LOBITS: 0
1202          003443              ADRCNT=LOBITS+1
1203
1204                                    ;SAVE PC OF TEST THAT FAILED AND R0-R5
1205                                    ;--------------------------------------
1206
1207  003444  016637  000004  001276 .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)
1208
1209                                    ;SAVE R0-R5
1210
1211  003452  010537  001272      SV05:   MOV     R5,SAVR5        ;SAVE R5
1212  003456  010437  001270              MOV     R4,SAVR4        ;SAVE R4
1213  003462  010337  001266              MOV     R3,SAVR3        ;SAVE R3
1214  003466  010237  001264              MOV     R2,SAVR2        ;SAVE R2
1215  003472  010137  001262              MOV     R1,SAVR1        ;SAVE R1
1216  003476  010037  001260              MOV     R0,SAVR0        ;SAVE R0
1217  003502  000002                      RTI                     ;LEAVE.
1218
1219                                    ;RESTORE R0-R5
1220
1221  003504  013700  001260      .RES05: MOV     SAVR0,R0        ;RESTORE R0
1222  003510  013701  001262              MOV     SAVR1,R1        ;RESTORE R1
1223  003514  013702  001264              MOV     SAVR2,R2        ;RESTORE R2
1224  003520  013703  001266              MOV     SAVR3,R3        ;RESTORE R3
1225  003524  013704  001270              MOV     SAVR4,R4        ;RESTORE R4
1226  003530  013705  001272              MOV     SAVR5,R5        ;RESTORE R5
```

```
1227  003534  000002                          RTI                     ;LEAVE
1228
1229                                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
1230                                          ;-----------------------------------------------------
1231
1232  003536  104402  005104          .CONVR: TYPE    ,MCRLF
1233  003542  010046                  .CNVRT: MOV     R0,-(SP)
1234  003544  010146                          MOV     R1,-(SP)
1235  003546  010346                          MOV     R3,-(SP)
1236  003550  010446                          MOV     R4,-(SP)
1237  003552  010546                          MOV     R5,-(SP)
1238  003554  017601  000012                  MOV     @12(SP),R1
1239  003560  062766  000002  000012          ADD     #2,12(SP)
1240  003566  012137  003742                  MOV     (R1)+,WRDCNT
1241  003572  112137  003744          1$:     MOVB    (R1)+,CHRCNT
1242  003576  112137  003745                  MOVB    (R1)+,SPACNT
1243  003602  013137  003746                  MOV     @(R1)+,BINWRD
1244  003606  013704  003746          2$:     MOV     BINWRD,R4
1245  003612  113705  003744                  MOVB    CHRCNT,R5
1246  003616  012700  005562                  MOV     #TEMP,R0
1247  003622  010403                  3$:     MOV     R4,R3
1248  003624  042703  177770                  BIC     #177770,R3
1249  003630  062703  000060                  ADD     #060,R3
1250  003634  110320                          MOVB    R3,(R0)+
1251  003636  000241                          CLC
1252  003640  006004                          ROR     R4
1253  003642  000241                          CLC
1254  003644  006004                          ROR     R4
1255  003646  000241                          CLC
1256  003650  006004                          ROR     R4
1257  003652  005305                          DEC     R5
1258  003654  001362                          BNE     3$
1259  003656  012703  005624                  MOV     #MDATA,R3
1260  003662  114023                  4$:     MOVB    -(R0),(R3)+
1261  003664  105337  003744                  DECB    CHRCNT
1262  003670  001374                          BNE     4$
1263  003672  105737  003745                  TSTB    SPACNT
1264  003676  001405                          BEQ     6$
1265  003700  112723  000040          5$:     MOVB    #040,(R3)+
1266  003704  105337  003745                  DECB    SPACNT
1267  003710  001373                          BNE     5$
1268  003712  105013                  6$:     CLRB    (R3)
1269  003714  104402  005624                  TYPE    ,MDATA
1270  003720  005337  003742                  DEC     WRDCNT
1271  003724  001322                          BNE     1$
1272  003726  012605                          MOV     (SP)+,R5
1273  003730  012604                          MOV     (SP)+,R4
1274  003732  012603                          MOV     (SP)+,R3
1275  003734  012601                          MOV     (SP)+,R1
1276  003736  012600                          MOV     (SP)+,R0
1277  003740  000002                          RTI
1278  003742  000000          WRDCNT: 0
1279  003744  000000          CHRCNT: 0
1280          003745          SPACNT=CHRCNT+1
1281  003746  000000          BINWRD: 0
1282
```

```
1283
1284                                    ;TRAP DISPATCH SERVICE
1285                                    ;ARGUMENT OF TRAP IS EXTRACTED
1286                                    ;AND USED AS OFFSET TO OBTAIN POINTER
1287                                    ;TO SELECTED SUBROUTINE
1288
1289  003750  011646            .TRPSR: MOV     (SP),-(SP)          ;GET PC OF RETURN
1290  003752  162716  000002            SUB     #2,(SP)             ;=PC OF TRAP
1291  003756  017616  000000            MOV     @(SP),(SP)          ;GET TRP
1292  003762  006316            TRPOK:  ASL     (SP)                ;MULTIPLY TRAP ARG BY 2
1293  003764  042716  177001            BIC     #177001,(SP)        ;CLEAR UNWANTED BITS
1294  003770  062716  001314            ADD     #.TRPTAB,(SP)       ;POINTER TO SUBROUTINE ADDRESS
1295  003774  017616  000000            MOV     @(SP),(SP)          ;SUBROUTINE ADDRESS
1296  004000  000136                    JMP     @(SP)+              ;GO TO SUBROUTINE
1297
1298                                    ;ERROR HANDLER
1299                                    ;-------------
1300
1301  004002                    .HLT:
1302  004002  022737  177570  001202    CMP     #177570,SWR     ;IS THERE A REAL SWR?
1303  004010  001411                    BEQ     64$             ;BR IF YES
1304  004012  017746  175170            MOV     @TKDBR,-(SP)    ;SAVE KEYBOARD CHAR
1305  004016  042716  000200            BIC     #BIT7,(SP)      ;CLEAR PARITY BIT
1306  004022  122726  000007            CMPB    #7,(SP)+        ;WAS IT CNTRL 'G' ?
1307  004026  001002                    BNE     .+6             ;BR IF NO.
1308  004030  004737  004640            JSR     PC,SERV.G       ;SERVICE ''CNTRL 'G'''.
1309  004034  032777  010000  175140 64$:  BIT   #SW12,@SWR      ;BELL ON ERROR?
1310  004042  001406                    BEQ     XBX             ;BR IF NO BELL
1311  004044  105777  175140            TSTB    @TPCSR          ;TTY READY.
1312  004050  100003                    BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
1313  004052  112777  000207  175132    MOVB    #207,@TPDBR     ;PUSH A BELL AT THE TTY.
1314  004060  032777  020000  175114 XBX:  BIT   #SW13,@SWR      ;DELETE ERROR PRINT OUT?
1315  004066  001105                    BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
1316  004070  021637  001234            CMP     (SP),LSTERR     ;WAS THIS ERROR FOUND LAST TIME?
1317  004074  001404                    BEQ     1$              ;BR IF YES
1318  004076  011637  001234            MOV     (SP),LSTERR     ;RECORD BEING HERE
1319  004102  105037  001311            CLRB    ERRFLG          ;PREPARE HEADER
1320  004106  104406            1$:     SAV05                   ;SAVE ALL PROC REGISTERS
1321  004110  011605                    MOV     (SP),R5         ;GET THE PC OF ERROR
1322  004112  162705  000002            SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
1323  004116  011504                    MOV     (R5),R4         ;GET HLT INSTRUCTION
1324  004120  006304                    ASL     R4              ;MULT BY TWO
1325  004122  061504                    ADD     (R5),R4         ;DOUBLE IT
1326  004124  006304                    ASL     R4              ;MULT AGAIN
1327  004126  042704  177001            BIC     #177001,R4      ;CLEAR JUNK
1328  004132  062704  033520            ADD     #.ERRTAB,R4     ;GET POINTER
1329  004136  012437  004252            MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
1330  004142  012437  004264            MOV     (R4)+,DATAHD    ;GET DATA HEADRER
1331  004146  011437  004276            MOV     (R4),DATABP     ;GET DATA TABLE
1332  004152  105737  001311          TSTB    ERRFLG          ;TYPE HEADREER
1333  004156  001403                    BEQ     TYPMSG          ;BR IF YES
1334  004160  005737  004276            TST     DATABP          ;DOES DATA TABLE EXIST?
1335  004164  001040                    BNE     TYPDAT          ;BR IF YES.
1336  004166  104402  005104 TYPMSG:  TYPE    ,MCRLF
1337  004172  104402  005104          TYPE    ,MCRLF
1338  004176  005737  001220          TST     LOCK
```

```
1339  004202  001402                        BEQ     1$
1340  004204  104402  005400                TYPE    ,MASTEK
1341  004210  104402  005366        1$:     TYPE    ,MTSTN
1342  004214  104411  004374                CNVRT   ,XTSTN          ;SHOW IT
1343  004220  104402  005454                TYPE    ,MERRPC         ;TYPE PC.
1344  004224  104411  004366                CNVRT   ,ERTABO         ;SHOW IT
1345  004230  104402  005104                TYPE    ,MCRLF          ;GIVE A CR/LF
1346  004234  112737  177777  001311        MOVB    #-1,ERRFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
1347  004242  005737  004252                TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
1348  004246  001402                        BEQ     WRKO.FM         ;BR IF NO.
1349  004250  104402                        TYPE                    ;TYPE
1350  004252  000000        ERRMSG: 0                               ;    ERROR MESSAGE
1351  004254                WRKO.FM:                                ;
1352  004254  005737  004264                TST     DATAHD          ;DATA HEADER?
1353  004260  001402                        BEQ     TYPDAT          ;BR IF NO
1354  004262  104402                        TYPE                    ;TYPE
1355  004264  000000        DATAHD: 0                               ;    DATA HEADER
1356  004266  005737  004276        TYPDAT: TST     DATABP          ;DATA TABLE?
1357  004272  001402                        BEQ     RESREG          ;BR IF NO.
1358  004274  104410                        CONVRT                  ;SHOW
1359  004276  000000        DATABP: 0                               ;    DATA TABLE
1360  004300  104407        RESREG: RES05                           ;RESTORE PROC REGISTERS
1361  004302  005777  174674        HALTS:  TST     @SWR            ;HALT ON ERROR?
1362  004306  100005                        BPL     EXITER          ;BR IF NO HALT ON ERROR
1363  004310  010046                        PUSHR0                  ;SAVE R0
1364  004312  016600  000002                MOV     2(SP),R0        ;SHOW ERROR PC IN DATA LIGHTS
1365  004316  000000                        HALT                    ;HALT
1366  004320  012600                        POPR0                   ;GET R0
1367  004322  005237  001232        EXITER: INC     ERRCNT          ;UPDATE ERROR COUNT
1368  004326  032777  000400  174646        BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
1369  004334  001007                        BNE     1$              ;BR IF YES
1370  004336  032777  002000  174636        BIT     #SW10,@SWR      ;GOTO NEXT TEST?
1371  004344  001407                        BEQ     2$              ;BR IF NO
1372  004346  013737  001216  001214        MOV     NEXT,RETURN     ;SET FOR NEXT TEST
1373  004354  012706  001200        1$:     MOV     #STACK,SP       ;RESET SP
1374  004360  000177  174630                JMP     @RETURN         ;GOTO SPECIFIED TEST
1375  004364  000002                2$:     RTI                     ;RETURN
1376  004366  000001        ERTABO: 1
1377  004370     006     002                .BYTE   6,2
1378  004372  001276                        SAVPC
1379  004374  000001        XTSTN:  1
1380  004376     003     002                .BYTE   3,2
1381  004400  001226                        TSTNO
1382                                        ;ENTER HERE ON POWER FAILURE
1383                                        ;--------------------------
1384
1385
1386  004402                .PFAIL:
1387  004402  012737  004414  000024        MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
1388  004410  000000                        HALT                            ;HALT ON POWER DOWN NORMAL
1389  004412  000777                        BR      .
1390
1391                                        ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
1392
1393  004414                RESTAR:
1394  004414  012737  004402  000024        MOV     #.PFAIL,24      ;SET UP FOR POWER FAILURE
```

```
 1395  004422  012706  001200                  MOV     #STACK,SP       ;RESET THE STACK POINTER
 1396  004426  005037  005562                  CLR     TEMP            ;READY FOR TIMMER
 1397  004432  005237  005562                  INC     TEMP            ;PLUS ONE TO THE TIMER!
 1398  004436  001375                          BNE     .-4             ;BR IF MORE TO GO
 1399  004440  104402  005107                  TYPE    ,MPFAIL         ;TYPE THE MESSAGE
 1400  004444  104411  004470                  CNVRT   ,PFTAB          ;TELL WHAT TEST TO RETURN TO.
 1401  004450  105037  001311                  CLRB    ERRFLG          ;START CLEAN
 1402  004454  005037  001234                  CLR     LSTERR          ;
 1403  004460  104412                          MSTCLR                  ;START CLEAN UP OF DEVICE
 1404  004462  104413                          RAMCLR                  ;CLEAR IT ALL!
 1405  004464  000177  174524                  JMP     @RETURN         ;START DOING THAT TEST AGAIN.
 1406  004470  000001                  PFTAB:  1
 1407  004472     003     002          .BYTE   3,2
 1408  004474  001226                          TSTNO
 1409  004476  010046                  .DELAY: MOV     R0,-(SP)
 1410  004500  013700  004514                  MOV     1$,R0
 1411  004504  005300                          DEC     R0
 1412  004506  001376                          BNE     .-2
 1413  004510  012600                          MOV     (SP)+,R0
 1414  004512  000002                          RTI
 1415  004514  000036                  1$:     30.
 1416
 1417  004516                          .RAMCLR:
 1418  004516  012777  004000  174636          MOV     #MRESET,@DVSCR  ;ISSUE A MASTER CLEAR
 1419  004524  010146                          MOV     R1,-(SP)        ;SAVE R1 ON THE STACK
 1420  004526  010446                          MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
 1421  004530  013701  001372                  MOV     DVSRS,R1        ;GET SECONDARY SEL. REG.
 1422  004534  013704  001376                  MOV     DVSRA,R4        ;GET SECONDARY REGISTER ACCESS REG.
 1423  004540  005014                  1$:     CLR     (R4)            ;ZERO THE SECONDARY REGISTER.
 1424  004542  062711  170361                  ADD     #^C<BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+BIT0,(R1)
 1425  004546  001374                          BNE     1$
 1426  004550  012604                          MOV     (SP)+,R4        ;RESTORE R4
 1427  004552  012601                          MOV     (SP)+,R1        ;RESTORE R1
 1428  004554  000002                          RTI
 1429
 1430  004556                          .MSTCLR:
 1431  004556  012777  004000  174576          MOV     #MRESET,@DVSCR  ;ISSUE MASTER CLEAR.
 1432  004564  000002                          RTI
 1433
 1434  004566                          .ROMCLK:
 1435  004566  052777  000002  174566          BIS     #BIT1,@DVSCR
 1436  004574  000002                          RTI
 1437
 1438  004576                          .DATACLK:
 1439  004576  010046                          MOV     R0,-(SP)
 1440  004600  005000                          CLR     R0
 1441  004602  052777  000400  174560          BIS     #BIT8,@DVLCR
 1442  004610  017737  174554  004636  1$:     MOV     @DVLCR,3$
 1443  004616  106037  004637                  RORB    3$+1
 1444  004622  103003                          BCC     2$
 1445  004624  005200                          INC     R0
 1446  004626  001370                          BNE     1$
 1447  004630  104000                          HLT     0
 1448  004632  012600                  2$:     MOV     (SP)+,R0
 1449  004634  000002                          RTI
 1450  004636  000001                  3$:     .BLKW 1
```

```
1451
1452  004640  032777  004000  174336  SERV.G: BIT     #4000,@TKCSR    ;RX BUSY?
1453  004646  001374                          BNE     SERV.G          ;BR IF YES
1454  004650  017737  174326  005072          MOV     @SWR,90$        ;SAVE (SWR).
1455  004656  013777  005072  174316  1$:     MOV     90$,@SWR        ;
1456  004664  104402  005052                  TYPE    ,89$            ;
1457  004670  104411  005064                  CNVRT   ,88$            ;
1458  004674  104402  005074                  TYPE    ,91$            ;
1459  004700  105777  174300                  TSTB    @TKCSR          ;WAIT FOR DONE.
1460  004704  100375                          BPL     .-4             ;
1461  004706  017746  174274                  MOV     @TKDBR,-(SP)    ;
1462  004712  042716  000200                  BIC     #BIT7,(SP)      ;
1463  004716  122726  000015                  CMPB    #15,(SP)+       ;
1464  004722  001450                          BEQ     5$              ;
1465  004724  005077  174252                  CLR     @SWR            ;
1466  004730  105777  174254          2$:     TSTB    @TPCSR          ;
1467  004734  100375                          BPL     .-4             ;
1468  004736  016677  177776  174246          MOV     -2(SP),@TPDBR   ;
1469  004744  000241                          CLC                     ;
1470  004746  006177  174230                  ROL     @SWR            ;
1471  004752  006177  174224                  ROL     @SWR            ;
1472  004756  006177  174220                  ROL     @SWR            ;
1473  004762  103735                          BCS     1$              ;ERROR
1474  004764  026627  177776  000060          CMP     -2(SP),#60      ;
1475  004772  002731                          BLT     1$              ;
1476  004774  026627  177776  000067          CMP     -2(SP),#67      ;
1477  005002  003325                          BGT     1$              ;
1478  005004  042766  177770  177776          BIC     #^C<7>,-2(SP)   ;
1479  005012  056677  177776  174162          BIS     -2(SP),@SWR     ;
1480  005020  105777  174160                  TSTB    @TKCSR          ;
1481  005024  100375                          BPL     .-4             ;
1482  005026  017746  174154                  MOV     @TKDBR,-(SP)
1483  005032  042716  000200                  BIC     #BIT7,(SP)
1484  005036  122726  000015                  CMPB    #15,(SP)+
1485  005042  001332                          BNE     2$              ;
1486  005044  104402  005104          5$:     TYPE    ,MCRLF          ;
1487  005050  000207                          RTS     PC
1488
1489  005052  020377  051450  051127  89$:     .ASCIZ  <377>? (SWR)=/?
1490  005060  036451  000057
1491                                  .EVEN
1492  005064  000001                  88$:     1
1493  005066     006     000                  .BYTE   6,0
1494  005070  005072                          90$
1495  005072  000000                  90$:     .WORD   0
1496  005074  036457  000057          91$:     .ASCIZ  ?/=/?
1497                                  .EVEN
1498  005100  020040  000077          MQM:     .ASCIZ  /  ?/
 (2)  005104  005015     000          MCRLF:   .ASCIZ   <15><12>
 (2)  005107     377  053520  020122  MPFAIL:  .ASCIZ  <377>/PWR FAILED. RESTART AT TEST /
 (2)  005145     377  047105  020104  MEPASS:  .ASCIZ  <377>/END PASS CZDVA-C  /
 (2)  005171     377  000122          MR:      .ASCIZ  <377>/R/
 (2)  005174  050377  047522  051107  MERR2:   .ASCIZ  <377>/PROGRAM INDICATES NO DEVICES PRESENT./
 (2)  005243     377  047111  052523  MERR3:   .ASCIZ  <377>/INSUFFICIENT DATA!/
 (2)  005267     377  042524  052123  MTSTPC:  .ASCIZ  <377>/TEST PC-/
 (2)  005301     377  047514  045503  MLOCK:   .ASCIZ  <377>/LOCK ON SELECTED TEST/
```

```
   (2)  005330  051503  035122  000040  MCSRX:   .ASCIZ   /CSR: /
   (2)  005336  042526  035103  000040  MVECX:   .ASCIZ  /VEC: /
   (2)  005344  040520  051523  051505  MPASSX:  .ASCIZ  /PASSES: /
   (2)  005355     105  051122  051117  MERRX:   .ASCIZ  /ERRORS: /
   (2)  005366  042524  052123  047040  MTSTN:   .ASCIZ  /TEST NO: /
   (2)  005400  000052                  MASTEK:  .ASCIZ  /*/
   (2)  005402  051777  052105  051440  MNEW:    .ASCIZ  <377>/SET SWITCH REG TO DV11'S DESIRED ACTIVE./
   (2)  005454  041520  020072     000  MERRPC:  .ASCIZ  /PC: /
   (2)  005461     377  040515  020120  XHEAD:   .ASCIZ  <377>/MAP OF DV11 STATUS/<377>
   (2)                                  .EVEN
   (2)  005506  000002                  XSTATQ:  2
1499    005510     006     003                   .BYTE   6,3
1500    005512  001246                          TEMP1
1501    005514     006     002                   .BYTE   6,2
1502    005516  001250                          TEMP2
1503                                 .EVEN
1504
1505                                          ;BUFFERS FOR INPUT-OUTPUT
1506
1507    005520  000000                  INBUF:   0
1508            005562                  .=.+40
1509    005562  000000                  TEMP:    0
1510            005624                  .=.+40
1511    005624  000000                  MDATA:   0
1512            005666                  .=.+40
```

```
1513
1514                                        ;
1515                                        ;ROUTINE USED TO "CYCLE" THROUGH UP TO EIGHT DV11'S
1516                                        ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
1517                                        ;AND RUNS THE SPECIFIED DV11'S.   THIS ROUTINE *MUST*
1518                                        ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
1519                                        ;SETUP NECESSARY.
1520                                        ;
1521
1522  005666  105737  001300        CYCLE: TSTB    DVACTV          ;ARE ANY DV11'S TO BE TESTED?
1523  005672  001004                       BNE     1$              ;BR IF OK.
1524  005674  104402  005174               TYPE    ,MERR2          ;NO DV11'S SELECTED!!
1525  005700  000000                       HALT                    ;STOP THE SHOW.
1526  005702  000776                       BR      .-2             ;DISQUALIFY CONT. SW.
1527  005704  133737  001304  001300  1$:  BITB    RUN,DVACTV      ;IS THIS ONE "ACTIVE"
1528  005712  001020                       BNE     2$              ;BR IF GOOD ONE FOUND.
1529  005714  000241                       CLC                     ;CLEAR PROC. CARRY BIT.
1530  005716  106137  001304               ROLB    RUN             ;UPDATE POINTER
1531  005722  105537  001304               ADCB    RUN             ;CATCH CARRY FROM RUN
1532  005726  062737  000024  001306       ADD     #24,CREAM       ;UPDATE ADDRESS POINTER.
1533  005734  022737  001740  001306       CMP     #DV.END,CREAM
1534  005742  001360                       BNE     1$              ;KEEP GOING; NOT ALL TESTED FOR.
1535  005744  012737  001500  001306       MOV     #DV.MAP,CREAM   ;RESET ADDRESS POINTER.
1536  005752  000754                       BR      1$              ;KEEP LOOKING FOR ACTIVE DV11
1537  005754  000241                 2$:   CLC                     ;CLEAR PROC. CARRY.
1538  005756  106137  001304               ROLB    RUN             ;UPDATE POINTER.
1539  005762  105537  001304               ADCB    RUN             ;CATCH CARRY.
1540  005766  013700  001306               MOV     CREAM,R0        ;GET ADDRESS POINTER.
1541  005772  062737  000024  001306       ADD     #24,CREAM       ;UPDATE.
1542  006000  022737  001740  001306       CMP     #DV.END,CREAM
1543                                                               ;ALL DONE?
1544  006006  001003                       BNE     3$              ;BR IF NO.
1545  006010  012737  001500  001306       MOV     #DV.MAP,CREAM   ;RESTORE POINTER.
1546  006016  012037  001362         3$:   MOV     (R0)+,DVSCR     ;LOAD SYSTEM CTRL. REG
1547  006022  012037  001352               MOV     (R0)+,DVRVEC    ;LOAD VECTOR
1548  006026  012037  001422               MOV     (R0)+,L00.03    ;GET LINE PARAMETERS. 00-03
1549  006032  012037  001432               MOV     (R0)+,SYNC2A    ;
1550  006036  012037  001424               MOV     (R0)+,L04.07    ;                     04-07
1551  006042  012037  001434               MOV     (R0)+,SYNC2B    ;
1552  006046  012037  001426               MOV     (R0)+,L08.11    ;                     08-11
1553  006052  012037  001436               MOV     (R0)+,SYNC2C    ;
1554  006056  012037  001430               MOV     (R0)+,L12.15    ;                     12-15
1555  006062  012037  001440               MOV     (R0)+,SYNC2D    ;
1556  006066  012700  000002               MOV     #2,R0           ;SAVE CORE THIS WAY!
1557  006072  013737  001362  001364       MOV     DVSCR,DVSCRH    ;GET SYS CTRL. REG HIGH BYTE.
1558  006100  005237  001364               INC     DVSCRH          ;GOT IT.
1559  006104  013737  001364  001366       MOV     DVSCRH,DVRIC    ;GET NXT REC. CHAR REG.
1560  006112  005237  001366               INC     DVRIC           ;GOT IT
1561  006116  013737  001366  001370       MOV     DVRIC,DVLCR     ;GET LN. PAR.REG.
1562  006124  060037  001370               ADD     R0,DVLCR        ;GOT IT
1563  006130  013737  001370  001372       MOV     DVLCR,DVSRS     ;GET SEC. REG. SEL. REG.
1564  006136  060037  001372               ADD     R0,DVSRS        ;GOT IT
1565  006142  013737  001372  001374       MOV     DVSRS,DVSRSH    ;GET HIGH BYTE.
1566  006150  005237  001374               INC     DVSRSH          ;GOT IT
1567  006154  013737  001374  001376       MOV     DVSRSH,DVSRA    ;SEC. REG. ACCESS.
1568  006162  005237  001376               INC     DVSRA           ;GOT IT
```

```
1569  006166  013737  001376  001400          MOV     DVSRA,DVSFR     ;SPEC. FUN. REG.
1570  006174  060037  001400                  ADD     R0,DVSFR        ;
1571  006200  013737  001400  001402          MOV     DVSFR,DVNSR     ;NPR STAT. REG.
1572  006206  060037  001402                  ADD     R0,DVNSR        ;
1573  006212  013737  001402  001404          MOV     DVNSR,RESV16    ;RESERVED REG
1574  006220  060037  001404                  ADD     R0,RESV16       ;
1575
1576  006224  013737  001352  001354          MOV     DVRVEC,DVRLVL   ;PTY LVL
1577  006232  060037  001354                  ADD     R0,DVRLVL       ;
1578  006236  013737  001354  001356          MOV     DVRLVL,DVTVEC   ;TX VEC
1579  006244  060037  001356                  ADD     R0,DVTVEC       ;
1580  006250  013737  001356  001360          MOV     DVTVEC,DVTLVL   ;TX LVL
1581  006256  060037  001360                  ADD     R0,DVTLVL
1582
1583  006262  012700  001422                  MOV     #L00.03,R0      ;LOAD STAUS 00-03
1584  006266  012701  001406                  MOV     #MASK.A,R1      ;PREPARE MASK.
1585  006272  012702  001416                  MOV     #CLK.A,R2       ;PREPARE CLOCKS
1586  006276  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.
1587
1588  006302  012700  001424                  MOV     #L04.07,R0      ;LOAD STAUS 00-03
1589  006306  012701  001410                  MOV     #MASK.B,R1      ;PREPARE MASK.
1590  006312  012702  001417                  MOV     #CLK.B,R2       ;PREPARE CLOCKS
1591  006316  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.
1592
1593  006322  012700  001426                  MOV     #L08.11,R0      ;LOAD STAUS 00-03
1594  006326  012701  001412                  MOV     #MASK.C,R1      ;PREPARE MASK.
1595  006332  012702  001420                  MOV     #CLK.C,R2       ;PREPARE CLOCKS
1596  006336  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.
1597
1598  006342  012700  001430                  MOV     #L12.15,R0      ;LOAD STAUS 00-03
1599  006346  012701  001414                  MOV     #MASK.D,R1      ;PREPARE MASK.
1600  006352  012702  001421                  MOV     #CLK.D,R2       ;PREPARE CLOCKS
1601  006356  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.
1602  006362  032777  000002  172612          BIT     #SW01,@SWR
1603  006370  001445                          BEQ     7$
1604  006372                          4$:
1605  006372  005737  000042                  TST     @#42
1606  006376  001042                          BNE     7$
1607  006400  104402  005104                  TYPE    ,MCRLF
1608  006404  104403                          INSTR
1609  006406  005366                          MTSTN
1610  006410  104405                          PARAM
1611  006412  000001                          1
1612  006414  001000                          1000
1613  006416  001226                          TSTNO
1614  006420     000                  .BYTE   0
1615  006421     001                  .BYTE   1
1616  006422  012700  007260                  MOV     #TST1,R0
1617  006426  022710                  5$:     CMP     (PC)+,(R0)
1618  006430  012737                          MOV     (PC)+,@(PC)+
1619  006432  001015                          BNE     6$
1620  006434  023760  001226  000002          CMP     TSTNO,2(R0)
1621  006442  001011                          BNE     6$
1622  006444  022760  001226  000004          CMP     #TSTNO,4(R0)
1623  006452  001005                          BNE     6$
1624  006454  010037  001214                  MOV     R0,RETURN
```

```
1625  006460  104402  005104                  TYPE    ,MCRLF
1626  006464  000412                          BR      8$
1627  006466  005720                  6$:     TST     (R0)+
1628  006470  020027  031354                  CMP     R0,#TLAST+10
1629  006474  001354                          BNE     5$
1630  006476  104402  005100                  TYPE    ,MQM
1631  006502  000733                          BR      4$
1632  006504  012737  007260  001214  7$:     MOV     #TST1,RETURN    ;PREPARE RETURN ADDRESS
1633  006512  000177  172476          8$:     JMP     @RETURN         ;GO START TESTING.
1634
1635  006516  011003                  FIX.00: MOV     (R0),R3         ;GET PARAMETERS.
1636  006520  042703  176377                  BIC     #^C<1400>,R3    ;CLEAR JUNK.
1637  006524  005703                          TST     R3              ;TEST FOR EIGHT BITS.
1638  006526  001005                          BNE     1$              ;BR IF NOT 8 BITS.
1639  006530  012711  000400                  MOV     #400,(R1)               ;SET FOR 8 BITS PER CHAR
1640  006534  112712  000010                  MOVB    #8.,(R2)        ;
1641  006540  000424                          BR      4$
1642  006542  022703  000400          1$:     CMP     #400,R3         ;CHECK FOR SEVEN BITS.
1643  006546  001005                          BNE     2$              ;BR IF NOT 7 BITS.
1644  006550  112711  000200                  MOVB    #200,(R1)       ;
1645  006554  112712  000007                  MOVB    #7,(R2)         ;
1646  006560  000414                          BR      4$
1647  006562  022703  001000          2$:     CMP     #1000,R3        ;CHECK FOR SIX BITS.
1648  006566  001005                          BNE     3$              ;BR IF NOT SIX BITS.
1649  006570  112711  000300                  MOVB    #300,(R1)       ;
1650  006574  112712  000006                  MOVB    #6,(R2)         ;
1651  006600  000404                          BR      4$
1652  006602  112711  000340          3$:     MOVB    #340,(R1)       ;IF NONE OF THE ABOVE; MUST BE 5 BITS.
1653  006606  112712  000005                  MOVB    #5,(R2)         ;
1654  006612  032710  040000          4$:     BIT     #PARBIT,(R0)    ;PARITY ENABLED?
1655  006616  001401                          BEQ     5$              ;IF =0; THEN NO PARITY.
1656  006620  105212                          INCB    (R2)            ;PLUS ONE TO THE CLOCK!
1657  006622  000207                  5$:     RTS     PC              ;
1658
1659                                          ;*ROUTINE USED TO ''AUTO SIZE'' THE DV11
1660                                          ;*CSR AND VECTOR.
1661                                          ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
1662                                          ;*      ADDRESS RANGE (175000:175400)
1663                                          ;*      AND THE VECTOR MAY BE ANY WHERE IN THE
1664                                          ;*      FLOATING VECTOR RANGE (300:770)
1665                                          ;*
1666
1667  006624                  AUTO.SIZE:
1668  006624  000005                          RESET                   ;INSURE A BUS INIT.
1669  006626  012702  001500  CSRMAP: MOV     #DV.MAP,R2      ;LOAD MAP POINTER.
1670  006632  005022          1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
1671  006634  022702  001740                  CMP     #DV.END,R2      ;ALL DONE?
1672  006640  001374                          BNE     1$              ;BR IF NO
1673  006642  105037  001301                  CLRB    DVNUM           ;SET OCTAL NUMBER OF DV11'S TO 0
1674  006646  012702  001500                  MOV     #DV.MAP,R2
1675  006652  012701  175000                  MOV     #175000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
1676  006656  012737  007076  000004          MOV     #6$,@#4         ;SET FOR NON-EXISTANT DEVICE TIME OUT
1677  006664  005711          2$:     TST     (R1)            ;IF DV11 DVSCR S/B 0
1678  006666  001037                          BNE     3$              ;IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO DV11
1679  006670  022761  177777  000012          CMP     #177777,12(R1)  ;IF DV11 THEN DVSFR S/B ALL 1'S ON INIT!
1680  006676  001033                          BNE     3$              ;BR IF NOT DV11
```

```
1681  006700  005761  000016                TST     16(R1)          ;IF DV11 THEN RESV16 S/B ALL 0'S
1682  006704  001030                      BNE     3$              ;BR IF NOT DV11
1683                                ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DV11 CSR ADDRESS.
1684  006706  010122                      MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
1685  006710  005722                      TST     (R2)+           ;POP OVER VECTOR STORE AREA
1686  006712  052722  000226              BIS     #226,(R2)+      ;SET LINE CARD 1 STAT AND SYNC
1687  006716  052722  000062              BIS     #62,(R2)+       ;
1688  006722  052722  000226              BIS     #226,(R2)+      ;SET LINE CARD 2 STAT AND SYNC
1689  006726  052722  000062              BIS     #62,(R2)+       ;
1690  006732  052722  000226              BIS     #226,(R2)+      ;SET LINE CARD 3 STAT AND SYNC
1691  006736  052722  000062              BIS     #62,(R2)+       ;
1692  006742  052722  000226              BIS     #226,(R2)+      ;SET LINE CARD 4 STAT AND SYNC
1693  006746  052722  000062              BIS     #62,(R2)+       ;
1694  006752  105237  001301              INCB    DVNUM           ;UPDATE DEVICE COUNTER
1695  006756  122737  000010  001301      CMPB    #10,DVNUM       ;ARE MAX. NO. OF DEV FOUND?
1696  006764  001405                      BEQ     100$            ;YES DON'T LOOK FOR ANY MORE.
1697  006766  062701  000010        3$:   ADD     #10,R1          ;UPDATE CSR POINTER ADDRESS
1698  006772  022701  175400              CMP     #175400,R1
1699  006776  001332                      BNE     2$              ;BR IF MORE ADDRESS TO CHECK.
1700  007000  012722  177777        100$: MOV     #177777,(R2)+   ;TERMINATER.
1701  007004  105037  001300              CLRB    DVACTV
1702  007010  105737  001301              TSTB    DVNUM           ;WERE ANY DV11'S FOUND AT ALL?
1703  007014  001423                      BEQ     5$              ;ERROR AUTO SIZER FOUND NO DV11'S IN THIS SYS.
1704  007016  113701  001301              MOVB    DVNUM,R1
1705  007022  110137  001303              MOVB    R1,SAVNUM       ;SAVE NUMBER OF DEVICES
1706  007026  000241                4$:   CLC
1707  007030  106137  001300              ROLB    DVACTV          ;GENERATE ACTIVE REGISTER OF DEVICES.
1708  007034  105237  001300              INCB    DVACTV          ;SET THE BIT
1709  007040  005301                      DEC     R1
1710  007042  001371                      BNE     4$              ;BR IF MORE TO GENERATE
1711  007044  012737  000006  000004      MOV     #6,@#4          ;RESTORE TRAP VECTOR
1712  007052  113737  001300  001302      MOVB    DVACTV,SAVACT   ;SAVE ACTIVE REGISTER
1713  007060  000137  007104              JMP     VECMAP          ;GO FIND THE VECTOR NOW.
1714  007064  104402  005174        5$:   TYPE    ,MERR2          ;NOTIFY OPR THAT NO DV11'S FOUND.
1715  007070  005000                      CLR     R0              ;MAKE DATA LIGHTS ZERO
1716  007072  000000                      HALT                    ;STOP THE SHOW
1717  007074  000776                      BR      .-2             ;DISABLE CONT. SW.
1718  007076  012716  006766        6$:   MOV     #3$,(SP)        ;ENTERED BY NON-EXISTANT TIME-OUT.
1719  007102  000002                      RTI                     ;RETURN TO MAINSTREAM
1720
1721  007104  012737  000340  000022 VECMAP: MOV  #340,@#22       ;SET IOT TRAP PRIO TO 7
1722  007112  012737  007234  000020      MOV     #4$,@#20        ;SET IOT TRAP VECTOR
1723  007120  012702  001500              MOV     #DV.MAP,R2      ;SET SOFTWARE POINTER
1724  007124  012700  000300              MOV     #300,R0         ;FLOATING VECTORS START HERE.
1725  007130  012701  000302              MOV     #302,R1         ;PC OF IOT INSTR.
1726  007134  010120                1$:   MOV     R1,(R0)+        ;START FILLING VECTOR AREA
1727  007136  012721  000004              MOV     #4,(R1)+        ;WITH .+2; IOT
1728  007142  022021                      CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
1729  007144  020127  001000              CMP     R1,#1000
1730  007150  101771                      BLOS    1$              ;BR IF MORE TO FILL
1731  007152  113737  001300  001246      MOVB    DVACTV,TEMP1    ;STORE TEMPORALLY
1732  007160  006037  001246        2$:   ROR     TEMP1           ;BRING OUT A BIT
1733  007164  103034                      BCC     5$              ;BR IF ALL DONE
1734  007166  005037  177776              CLR     PS              ;ZERO CPU PRIO
1735  007172  012772  001300  000000      MOV     #BIT9+BIT7+BIT6,@(R2)
1736  007200  005000                      CLR     R0              ;ATTEMPT TO FORCE AN INTERUPT
```

```
 1737  007202  005200                        INC     R0              ;STALL
 1738  007204  001376                        BNE     .-2             ;       FOR TIME TO INTERUPT
 1739  007206  052762  000300  000002        BIS     #300,2(R2)      ;NO INTERUPT ASSUME 300 AND FIX DV11 LATER
 1740  007214  042772  176777  000000  3$:   BIC     #^C<BIT9>,@(R2)
 1741  007222  005072  000000                CLR     @(R2)
 1742  007226  062702  000024                ADD     #24,R2          ;POP SOFTWARE POINTER
 1743  007232  000752                        BR      2$              ;KEEP GOING
 1744  007234  051662  000002          4$:   BIS     (SP),2(R2)      ;GET VECTOR ADDRESS
 1745  007240  042762  000007  000002        BIC     #7,2(R2)        ;CLEAR JUNK
 1746  007246  022626                        CMP     (SP)+,(SP)+     ;POP IOT JUNK OFF STACK
 1747  007250  012716  007214                MOV     #3$,(SP)        ;SET FOR RETURN
 1748  007254  000002                        RTI
 1749  007256  000207                  5$:   RTS     PC              ;ALL DONE WITH ''AUTO SIZING''
 1750
```

```
                                        ;************************ TEST 1 *******************************
                                        ;*VERIFY THAT ADDRESSING DEVICE DOES *NOT* CAUSE
                                        ;*A TIME-OUT TRAP.
                                        ;:***************************************************************

                                ;  TEST 1
                                ;--------------
007260  012737  000001  001226  TST1:   MOV     #1,TSTNO
007266  012737  007370  001216          MOV     #TST2,NEXT
007274  012737  007326  001220          MOV     #1$,LOCK
007302  012700  000010                  MOV     #8.,R0          ;SET FOR MAX. 8 PRI. REGISTERS
007306  013701  001362                  MOV     DVSCR,R1        ;GET FIRST PRI. ADDRESS
007312  012737  007362  000004          MOV     #2$,4           ;SET FOR TIME-OUT TRAP.
007320  012737  000340  000006          MOV     #340,6          ;SAFE GUARD.
007326  005711                  1$:     TST     (R1)            ;REFERENCE THE ADDRESS.
007330  000240                          NOP                     ;STALL
007332  000240                          NOP                     ;     FOR TIME.
007334  104401                          SCOP1                   ;IF SW09=1; GOTO 1$
007336  062701  000002                  ADD     #2,R1           ;UPDATE TO NEXT ADDRESS.
007342  005300                          DEC     R0              ;ARE ALL ADDRESS CHECKED?
007344  001370                          BNE     1$              ;BR IF NO.
007346  012737  000006  000004          MOV     #6,4            ;RESET TRAP ZONE.
007354  005037  000006                  CLR     @#6             ;
007360  104400                          SCOPE                   ;SCOPE THIS TEST
007362  011602                  2$:     MOV     (SP),R2         ;SAVE THE TRAP PC
007364  104001                          HLT     1               ;REPORT TIME-OUT TRAP
007366  000002                          RTI                     ;RETURN TO MAIN PROGRAM


                                        ;************************ TEST 2 *******************************
                                        ;*PRIMARY REGISTER ADDRESSING TEST
                                        ;*LOAD EACH PRIMARY REGISTER WITH A
                                        ;*DIFFERENT NUMBER AND VERIFY EACH
                                        ;*WAS INDIVIDUALLY ADDRESSED.
                                        ;:***************************************************************

                                ;  TEST 2
                                ;--------------
007370  012737  000002  001226  TST2:   MOV     #2,TSTNO
007376  012737  007622  001216          MOV     #TST3,NEXT
007404  012737  007440  001220          MOV     #1$,LOCK
007412  012700  007602                  MOV     #3$,R0          ;SET DATA TABLE POINTER
007416  013703  001362                  MOV     DVSCR,R3        ;SET DV POINTER
007422  005013                          CLR     (R3)            ;START REG AT ZERO
007424  005077  171742                  CLR     @DVSRS          ;ZERO SEC. REG SEL.
007430  005077  171742                  CLR     @DVSRA          ;ZERO SEC REG ACCESS
007434  012702  000010                  MOV     #8.,R2          ;SET FOR EIGHT PRIMARY REGISTERS.
007440  011005                  1$:     MOV     (R0),R5         ;PUT DATA INTO EXPECTED
007442  010513                          MOV     R5,(R3)         ;WRITE EXPECTED INTO DV REGISTER
007444  011304                          MOV     (R3),R4         ;READ REGISTER INTO FOUND LOC
007446  020504                          CMP     R5,R4           ;DOES EXPECTED=RECEIVED?
007450  001401                          BEQ     64$             ;BR IF YES
007452  104003                          HLT     3               ;THIS IS A DATA ERROR *NOT* A DUEL ADDRESSING ERROR. NOT
007454  104401                  64$:    SCOP1                   ;SW09=1?
007456  022023                          CMP     (R0)+,(R3)+     ;POP DATA POINTERS AND HRDW POINTER
007460  020337  001402                  CMP     R3,DVNSR        ;DON'T DO THE DVNSR!
```

```
1807  007464  001002                        BNE     4$              ;BR IF NOT DVNSR
1808  007466  022320                        CMP     (R3)+,(R0)+     ;POP POINTERS AROUND DVNSR
1809  007470  005302                        DEC     R2              ;UPDATE REGISTER COUNTER
1810  007472  020337  001370          4$:   CMP     R3,DVLCR        ;DON'T DO THE DVLCR!
1811  007476  001002                        BNE     6$              ;BR IF NOT THE DVLCR
1812  007500  022320                        CMP     (R3)+,(R0)+     ;POP POINTERS AROUND THE DVLCR
1813  007502  005302                        DEC     R2              ;UPDATE THE REGISTER COUNTER
1814  007504  005302                  6$:   DEC     R2              ;DITTO
1815  007506  001354                        BNE     1$              ;BR IF MORE TO GO
1816                                  ;CHECK DUEL ADDRESSING......
1817  007510  012700  007602                MOV     #3$,R0          ;SET DATA POINTER
1818  007514  013703  001362                MOV     DVSCR,R3        ;SET HRDW POINTER
1819  007520  012737  007532  001220        MOV     #2$,LOCK        ;SET IF SW09=1
1820  007526  012702  000010                MOV     #8.,R2          ;SET EIGHT PRIMARY REGISTERS
1821  007532  011005                  2$:   MOV     (R0),R5         ;LOAD DATA INTO EXPECTED
1822  007534  011304                        MOV     (R3),R4         ;READ THE DV REGISTER
1823  007536  020504                        CMP     R5,R4           ;DOES THE DATA COMPARE
1824  007540  001401                        BEQ     65$             ;BR IF OK
1825  007542  104003                        HLT     3               ;NOW THIS WAS A DUEL ADDRESSING ERROR.
1826  007544  104401                  65$:  SCOP1                   ;SW09=1?
1827  007546  022023                        CMP     (R0)+,(R3)+     ;POP POINTERS
1828  007550  020337  001402                CMP     R3,DVNSR        ;DON'T DO THE DVNSR
1829  007554  001002                        BNE     5$              ;BR IF NOT DVNSR
1830  007556  022320                        CMP     (R3)+,(R0)+     ;POP POINTERS
1831  007560  005302                        DEC     R2              ;SET REG COUNTER
1832  007562  020337  001370          5$:   CMP     R3,DVLCR        ;DON'T DO THE DVLCR
1833  007566  001002                        BNE     7$              ;BR IF NOT DVLCR
1834  007570  022320                        CMP     (R3)+,(R0)+     ;POP POINTERS
1835  007572  005302                        DEC     R2              ;SET REG POINTER
1836  007574  005302                  7$:   DEC     R2              ;DITTO
1837  007576  001355                        BNE     2$              ;BR IF MORE TO GO
1838  007600  104400                        SCOPE                   ;SCOPE THIS TEST
1839  007602  000010                  3$:   .WORD   000010          ;DVSCR
1840  007604  000000                        .WORD   000000          ;DVRIC
1841  007606  000000                        .WORD   SKIP            ;DVLCR
1842  007610  001400                        .WORD   001400          ;DVSRS
1843  007612  000300                        .WORD   000300          ;DVSRA
1844  007614  100000                        .WORD   100000          ;DVSFR
1845  007616  000000                        .WORD   SKIP            ;DVNSR
1846  007620  000060                        .WORD   000060          ;RESV16
1847
1848
1849                                  ;************************* TEST 3 *******************************
1850                                  ;*SYSTEM CONTROL REGSTER READ/WRITE TEST.
1851                                  ;*SET BIT2, VERIFY BIT2 WAS SET.
1852                                  ;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.
1853                                  ;:*************************************************************************
1854
1855                                  ;  TEST 3
1856                                  ;--------------
1857  007622  012737  000003  001226  TST3:  MOV     #3,TSTNO
1858  007630  012737  007676  001216        MOV     #TST4,NEXT
1859  007636  013703  001362                MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
1860  007642  012705  000004                MOV     #BIT2,R5        ;SET 'EXPECTED '.
1861  007646  010513                        MOV     R5,(R3)         ;WRITE THE REGISTER.
1862  007650  011304                        MOV     (R3),R4         ;READ THE REGISTER.
```

```
1863  007652  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
1864  007654  001401                          BEQ     1$              ;ARE THEY THE SAME?
1865  007656  104003                          HLT     3               ;COMPARISON ERROR.
1866  007660  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT2
1867  007662  011304                          MOV     (R3),R4         ;READ THE REGISTER.
1868  007664  005005                          CLR     R5              ;SET 'EXPECTED''
1869  007666  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
1870  007670  001401                          BEQ     2$              ;BR IF OK
1871  007672  104003                          HLT     3               ;COMPARISON ERROR
1872  007674  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
1873
1874
1875                                          ;************************ TEST 4 ******************************
1876                                          ;*SYSTEM CONTROL REGSTER READ/WRITE TEST.
1877                                          ;*SET BIT3, VERIFY BIT3 WAS SET.
1878                                          ;*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
1879                                          ;:**************************************************************************
1880
1881                                  ;  TEST 4
1882                                  ;--------------
1883  007676  012737  000004  001226  TST4:   MOV     #4,TSTNO
1884  007704  012737  007752  001216          MOV     #TST5,NEXT
1885  007712  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
1886  007716  012705  000010                  MOV     #BIT3,R5        ;SET 'EXPECTED ''.
1887  007722  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
1888  007724  011304                          MOV     (R3),R4         ;READ THE REGISTER.
1889  007726  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
1890  007730  001401                          BEQ     1$              ;ARE THEY THE SAME?
1891  007732  104003                          HLT     3               ;COMPARISON ERROR.
1892  007734  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT3
1893  007736  011304                          MOV     (R3),R4         ;READ THE REGISTER.
1894  007740  005005                          CLR     R5              ;SET 'EXPECTED''
1895  007742  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
1896  007744  001401                          BEQ     2$              ;BR IF OK
1897  007746  104003                          HLT     3               ;COMPARISON ERROR
1898  007750  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
1899
1900
1901                                          ;************************ TEST 5 ******************************
1902                                          ;*SYSTEM CONTROL REGSTER READ/WRITE TEST.
1903                                          ;*SET BIT4, VERIFY BIT4 WAS SET.
1904                                          ;*CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
1905                                          ;:**************************************************************************
1906
1907                                  ;  TEST 5
1908                                  ;--------------
1909  007752  012737  000005  001226  TST5:   MOV     #5,TSTNO
1910  007760  012737  010026  001216          MOV     #TST6,NEXT
1911  007766  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
1912  007772  012705  000020                  MOV     #BIT4,R5        ;SET 'EXPECTED ''.
1913  007776  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
1914  010000  011304                          MOV     (R3),R4         ;READ THE REGISTER.
1915  010002  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
1916  010004  001401                          BEQ     1$              ;ARE THEY THE SAME?
1917  010006  104003                          HLT     3               ;COMPARISON ERROR.
1918  010010  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT4
```

```
                        MOV     (R3),R4         ;READ THE REGISTER.
                        CLR     R5              ;SET 'EXPECTED''
                        CMP     R5,R4           ;R5=GOOD; R4=?
                        BEQ     2$              ;BR IF OK
                        HLT     3               ;COMPARISON ERROR
                2$:     SCOPE                   ;SCOPE THIS TEST

                        ;************************ TEST 6 ******************************
                        ;*SYSTEM CONTROL REGSTER READ/WRITE TEST.
                        ;*SET BIT5, VERIFY BIT5 WAS SET.
                        ;*CLEAR BIT5, VERIFY BIT5 WAS CLEARED.
                        ;:**************************************************************

                ;  TEST 6
                ;--------------
                TST6:   MOV     #6,TSTNO
                        MOV     #TST7,NEXT
                        MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
                        MOV     #BIT5,R5        ;SET 'EXPECTED ''.
                        MOV     R5,(R3)         ;WRITE THE REGISTER.
                        MOV     (R3),R4         ;READ THE REGISTER.
                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
                        BEQ     1$              ;ARE THEY THE SAME?
                        HLT     3               ;COMPARISON ERROR.
                1$:     BIC     R5,(R3)         ;CLEAR BIT5
                        MOV     (R3),R4         ;READ THE REGISTER.
                        CLR     R5              ;SET 'EXPECTED''
                        CMP     R5,R4           ;R5=GOOD; R4=?
                        BEQ     2$              ;BR IF OK
                        HLT     3               ;COMPARISON ERROR
                2$:     SCOPE                   ;SCOPE THIS TEST

                        ;************************ TEST 7 ******************************
                        ;*SYSTEM CONTROL REGSTER READ/WRITE TEST.
                        ;*SET BIT6, VERIFY BIT6 WAS SET.
                        ;*CLEAR BIT6, VERIFY BIT6 WAS CLEARED.
                        ;:**************************************************************

                ;  TEST 7
                ;--------------
                TST7:   MOV     #7,TSTNO
                        MOV     #TST10,NEXT
                        MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
                        MOV     #BIT6,R5        ;SET 'EXPECTED ''.
                        MOV     R5,(R3)         ;WRITE THE REGISTER.
                        MOV     (R3),R4         ;READ THE REGISTER.
                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
                        BEQ     1$              ;ARE THEY THE SAME?
                        HLT     3               ;COMPARISON ERROR.
                1$:     BIC     R5,(R3)         ;CLEAR BIT6
                        MOV     (R3),R4         ;READ THE REGISTER.
                        CLR     R5              ;SET 'EXPECTED''
                        CMP     R5,R4           ;R5=GOOD; R4=?
                        BEQ     2$              ;BR IF OK
```

```
                                        HLT     3               ;COMPARISON ERROR
                                2$:     SCOPE                   ;SCOPE THIS TEST

                                        ;************************ TEST 10 ****************************
                                        ;*TEST THAT BIT 7(RECEIVER INTERUPT)
                                        ;*CANNOT BE WRITTEN WHEN BIT 9 (SYSTEM MAINTAINCE)
                                        ;*IS NOT SET.
                                        ;*THEN VERIFY THAT BIT 7 CAN BE WRITTEN WHEN
                                        ;*BIT 9 IS SET.
                                        ;:**********************************************************

                                ;  TEST 10
                                ;--------------
001226  012737  000010  001226  TST10:  MOV     #10,TSTNO
001216  012737  010252  001216          MOV     #TST11,NEXT
001362  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER POINTER INTO R3
        005005                          CLR     R5              ;SET EXPECTED RESULT
000200  012713  000200                  MOV     #BIT7,(R3)      ;ATTEMPT TO SET BIT 7
        011304                          MOV     (R3),R4         ;READ BACK REGISTER
        001401                          BEQ     1$              ;BIT 7 SHOULD NOT BE SET
        104003                          HLT     3               ;DVSCR NOT ALL 0'S
001200  012705  001200          1$:     MOV     #BIT7+BIT9,R5   ;SET BIT 7+BIT9 IN THE DVSCR
        010513                          MOV     R5,(R3)         ;
        011304                          MOV     (R3),R4         ;READ REGISTER
        020504                          CMP     R5,R4           ;IS BIT 9 AND BIT 7 SET?
        001401                          BEQ     2$              ;BR IF OK
        104003                          HLT     3               ;DVSCR ERROR
000200  042705  000200          2$:     BIC     #BIT7,R5        ;CLEAR BIT 7
000200  042713  000200                  BIC     #BIT7,(R3)      ;DITTO
        011304                          MOV     (R3),R4         ;READ REGISTER
        020504                          CMP     R5,R4           ;COMPARE OK?
        001401                          BEQ     3$              ;BR IF YES
        104003                          HLT     3               ;DVSCR ERROR
        104400                  3$:     SCOPE                   ;SCOPE THIS TEST


                                        ;************************ TEST 11 ****************************
                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT8, VERIFY BIT8 WAS SET.
                                        ;*CLEAR BIT8, VERIFY BIT8 WAS CLEARED.
                                        ;:**********************************************************

                                ;  TEST 11
                                ;--------------
001226  012737  000011  001226  TST11:  MOV     #11,TSTNO
001216  012737  010326  001216          MOV     #TST12,NEXT
001362  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
000400  012705  000400                  MOV     #BIT8,R5        ;SET 'EXPECTED ''.
        010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
        011304                          MOV     (R3),R4         ;READ THE REGISTER.
        020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
        001401                          BEQ     1$              ;ARE THEY THE SAME?
        104003                          HLT     3               ;COMPARISON ERROR.
        040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT8
        011304                          MOV     (R3),R4         ;READ THE REGISTER.
```

```
2031  010314  005005                        CLR     R5              ;SET 'EXPECTED''
2032  010316  020504                        CMP     R5,R4           ;R5=GOOD; R4=?
2033  010320  001401                        BEQ     2$              ;BR IF OK
2034  010322  104003                        HLT     3               ;COMPARISON ERROR
2035  010324  104400                2$:     SCOPE                   ;SCOPE THIS TEST
2036
2037
2038                                        ;************************ TEST 12 ******************************
2039                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
2040                                        ;*SET BIT9, VERIFY BIT9 WAS SET.
2041                                        ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED.
2042                                        ;:**************************************************************************
2043
2044                                ;  TEST 12
2045                                ;--------------
2046  010326  012737  000012  001226 TST12:  MOV     #12,TSTNO
2047  010334  012737  010402  001216         MOV     #TST13,NEXT
2048  010342  013703  001362                 MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
2049  010346  012705  001000                 MOV     #BIT9,R5        ;SET 'EXPECTED ''.
2050  010352  010513                         MOV     R5,(R3)         ;WRITE THE REGISTER.
2051  010354  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2052  010356  020504                         CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2053  010360  001401                         BEQ     1$              ;ARE THEY THE SAME?
2054  010362  104003                         HLT     3               ;COMPARISON ERROR.
2055  010364  040513                1$:      BIC     R5,(R3)         ;CLEAR BIT9
2056  010366  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2057  010370  005005                         CLR     R5              ;SET 'EXPECTED''
2058  010372  020504                         CMP     R5,R4           ;R5=GOOD; R4=?
2059  010374  001401                         BEQ     2$              ;BR IF OK
2060  010376  104003                         HLT     3               ;COMPARISON ERROR
2061  010400  104400                2$:     SCOPE                   ;SCOPE THIS TEST
2062
2063
2064                                        ;************************ TEST 13 ******************************
2065                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
2066                                        ;*SET BIT10, VERIFY BIT10 WAS SET.
2067                                        ;*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.
2068                                        ;:**************************************************************************
2069
2070                                ;  TEST 13
2071                                ;--------------
2072  010402  012737  000013  001226 TST13:  MOV     #13,TSTNO
2073  010410  012737  010456  001216         MOV     #TST14,NEXT
2074  010416  013703  001362                 MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
2075  010422  012705  002000                 MOV     #BIT10,R5       ;SET 'EXPECTED ''.
2076  010426  010513                         MOV     R5,(R3)         ;WRITE THE REGISTER.
2077  010430  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2078  010432  020504                         CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2079  010434  001401                         BEQ     1$              ;ARE THEY THE SAME?
2080  010436  104003                         HLT     3               ;COMPARISON ERROR.
2081  010440  040513                1$:      BIC     R5,(R3)         ;CLEAR BIT10
2082  010442  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2083  010444  005005                         CLR     R5              ;SET 'EXPECTED''
2084  010446  020504                         CMP     R5,R4           ;R5=GOOD; R4=?
2085  010450  001401                         BEQ     2$              ;BR IF OK
2086  010452  104003                         HLT     3               ;COMPARISON ERROR
```

```
2087  010454  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
2088
2089
2090                                          ;************************ TEST 14 ******************************
2091                                          ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
2092                                          ;*SET BIT12, VERIFY BIT12 WAS SET.
2093                                          ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED.
2094                                          ;;*************************************************************************
2095
2096                                  ;  TEST 14
2097                                  ;--------------
2098  010456  012737  000014  001226  TST14:  MOV     #14,TSTNO
2099  010464  012737  010532  001216          MOV     #TST15,NEXT
2100  010472  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
2101  010476  012705  010000                  MOV     #BIT12,R5       ;SET 'EXPECTED ''.
2102  010502  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
2103  010504  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2104  010506  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2105  010510  001401                          BEQ     1$              ;ARE THEY THE SAME?
2106  010512  104003                          HLT     3               ;COMPARISON ERROR.
2107  010514  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT12
2108  010516  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2109  010520  005005                          CLR     R5              ;SET 'EXPECTED''
2110  010522  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
2111  010524  001401                          BEQ     2$              ;BR IF OK
2112  010526  104003                          HLT     3               ;COMPARISON ERROR
2113  010530  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
2114
2115
2116                                          ;************************ TEST 15 ******************************
2117                                          ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
2118                                          ;*SET BIT13, VERIFY BIT13 WAS SET.
2119                                          ;*CLEAR BIT13, VERIFY BIT13 WAS CLEARED.
2120                                          ;;*************************************************************************
2121
2122                                  ;  TEST 15
2123                                  ;--------------
2124  010532  012737  000015  001226  TST15:  MOV     #15,TSTNO
2125  010540  012737  010606  001216          MOV     #TST16,NEXT
2126  010546  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
2127  010552  012705  020000                  MOV     #BIT13,R5       ;SET 'EXPECTED ''.
2128  010556  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
2129  010560  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2130  010562  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2131  010564  001401                          BEQ     1$              ;ARE THEY THE SAME?
2132  010566  104003                          HLT     3               ;COMPARISON ERROR.
2133  010570  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT13
2134  010572  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2135  010574  005005                          CLR     R5              ;SET 'EXPECTED''
2136  010576  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
2137  010600  001401                          BEQ     2$              ;BR IF OK
2138  010602  104003                          HLT     3               ;COMPARISON ERROR
2139  010604  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
2140
2141
2142                                          ;************************ TEST 16 ******************************
```

```
                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT3+BIT0, VERIFY BIT3+BIT0 WAS SET.
                                        ;*CLEAR BIT3+BIT0, VERIFY BIT3+BIT0 WAS CLEARED.
                                        ;;*****************************************************************

                                ;  TEST 16
                                ;--------------
                                ;
010606  012737  000016  001226  TST16:  MOV     #16,TSTNO
010614  012737  010662  001216          MOV     #TST17,NEXT
010622  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
010626  012705  000011                  MOV     #BIT3+BIT0,R5   ;SET 'EXPECTED ''.
010632  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
010634  011304                          MOV     (R3),R4         ;READ THE REGISTER.
010636  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
010640  001401                          BEQ     1$              ;ARE THEY THE SAME?
010642  104003                          HLT     3               ;COMPARISON ERROR.
010644  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT3+BIT0
010646  011304                          MOV     (R3),R4         ;READ THE REGISTER.
010650  005005                          CLR     R5              ;SET 'EXPECTED''
010652  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
010654  001401                          BEQ     2$              ;BR IF OK
010656  104003                          HLT     3               ;COMPARISON ERROR
010660  104400                  2$:     SCOPE                   ;SCOPE THIS TEST

                                        ;************************* TEST 17 *******************************
                                        ;*TEST THAT BIT 15(NPR STATUS OVERFLOW INTERUPT)
                                        ;*CANNOT BE WRITTEN WHEN BIT 9 (SYSTEM MAINTAINCE)
                                        ;*IS NOT SET.
                                        ;*THEN VERIFY THAT BIT 15 CAN BE WRITTEN WHEN
                                        ;*BIT 9 IS SET.
                                        ;;*****************************************************************

                                ;  TEST 17
                                ;--------------
                                ;
010662  012737  000017  001226  TST17:  MOV     #17,TSTNO
010670  012737  010756  001216          MOV     #TST20,NEXT
010676  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER POINTER INTO R3
010702  005005                          CLR     R5              ;SET EXPECTED RESULT
010704  012713  100000                  MOV     #BIT15,(R3)     ;ATTEMPT TO SET BIT 15
010710  011304                          MOV     (R3),R4         ;READ BACK REGISTER
010712  001401                          BEQ     1$              ;BIT 15 SHOULD NOT BE SET
010714  104003                          HLT     3               ;DVSCR NOT ALL 0'S
010716  012705  101000          1$:     MOV     #BIT15+BIT9,R5  ;SET BIT 15+BIT9 IN THE DVSCR
010722  010513                          MOV     R5,(R3)         ;
010724  011304                          MOV     (R3),R4         ;READ REGISTER
010726  020504                          CMP     R5,R4           ;IS BIT 9 AND BIT 15 SET?
010730  001401                          BEQ     2$              ;BR IF OK
010732  104003                          HLT     3               ;DVSCR ERROR
010734  042705  100000          2$:     BIC     #BIT15,R5       ;CLEAR BIT 15
010740  042713  100000                  BIC     #BIT15,(R3)     ;DITTO
010744  011304                          MOV     (R3),R4         ;READ REGISTER
010746  020504                          CMP     R5,R4           ;COMPARE OK?
010750  001401                          BEQ     3$              ;BR IF YES
010752  104003                          HLT     3               ;DVSCR ERROR
010754  104400                  3$:     SCOPE                   ;SCOPE THIS TEST
```

```
2199
2200                                    ;************************ TEST 20 *****************************
2201                                    ;*TEST THAT BIT8 IN DVSCR CLEARS
2202                                    ;*BIT7 OF DVSCR.
2203                                    ;;*************************************************************************
2204
2205                                ;  TEST 20
2206                                ;--------------
2207  010756  012737  000020  001226  TST20:  MOV     #20,TSTNO
2208  010764  012737  011034  001216          MOV     #TST21,NEXT
2209  010772  013703  001362                  MOV     DVSCR,R3        ;GET POINTER
2210  010776  012705  000400                  MOV     #BIT8,R5        ;GET GOOD
2211  011002  012713  000200                  MOV     #BIT7,(R3)      ;SET BIT7
2212  011006  052713  000400                  BIS     #BIT8,(R3)      ;SET BIT8
2213  011012  027777  170350  170346          CMP     @DVRIC,@DVRIC   ;WASTE TIME
2214  011020  011304                          MOV     (R3),R4         ;READ SCR
2215  011022  020504                          CMP     R5,R4           ;BIT8 ONLY SET?
2216  011024  001401                          BEQ     1$              ;BR IF YES
2217  011026  104003                          HLT     3               ;BIT8 NOT ONLY BIT SET OR 8 ISN'T SET!
2218  011030  005013                  1$:     CLR     (R3)            ;LEAVE REG=0
2219  011032  104400                          SCOPE                   ;SCOPE
2220
2221
2222                                    ;************************ TEST 21 *****************************
2223                                    ;*RECEIVER INTERUPT CHARACTER REGISTER TEST
2224                                    ;*TEST THAT RECEIVER INTERUPT CHARACTER REGISTER CANNOT BE WRITTEN.
2225                                    ;*WRITE RECEIVER INTERUPT CHARACTER REGISTER WITH ALL 1'S
2226                                    ;*AND VERIFY THAT ALL 0'S ARE READ BACK.
2227                                    ;;*************************************************************************
2228
2229                                ;  TEST 21
2230                                ;--------------
2231  011034  012737  000021  001226  TST21:  MOV     #21,TSTNO
2232  011042  012737  011074  001216          MOV     #TST22,NEXT
2233  011050  013703  001366                  MOV     DVRIC,R3
2234                                                                  ;GET THE REGISTER.
2235  011054  005005                          CLR     R5              ;SET EXPECTED (ZERO)
2236  011056  012713  177777                  MOV     #-1,(R3)        ;WRITE REGISTER WITH ALL 1'S
2237  011062  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2238  011064  020504                          CMP     R5,R4           ;IS THE REGISTER EQUAL TO ZERO.
2239  011066  001401                          BEQ     1$              ;BR IF OK.
2240  011070  104003                          HLT     3               ;REGISTER NOT ZERO.
2241  011072  104400                  1$:     SCOPE                   ;SCOPE THIS TEST.
2242
2243
2244                                    ;************************ TEST 22 *****************************
2245                                    ;*LINE CONTROL REGISTER READ/WRITE TEST.
2246                                    ;*SET BIT9, VERIFY BIT9 WAS SET.
2247                                    ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED.
2248                                    ;;*************************************************************************
2249
2250                                ;  TEST 22
2251                                ;--------------
2252  011074  012737  000022  001226  TST22:  MOV     #22,TSTNO
2253  011102  012737  011160  001216          MOV     #TST23,NEXT
2254  011110  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
```

```
2255  011114  012705  001000              MOV     #BIT9,R5          ;SET 'EXPECTED ''.
2256  011120  010513                      MOV     R5,(R3)           ;WRITE THE REGISTER.
2257  011122  011304                      MOV     (R3),R4           ;READ THE REGISTER.
2258  011124  042704  000063              BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
2259  011130  020504                      CMP     R5,R4             ;R5=GOOD; R4=UNKNOWN.
2260  011132  001401                      BEQ     1$                ;ARE THEY THE SAME?
2261  011134  104003                      HLT     3                 ;COMPARISON ERROR.
2262  011136  040513              1$:     BIC     R5,(R3)           ;CLEAR BIT9
2263  011140  011304                      MOV     (R3),R4           ;READ THE REGISTER.
2264  011142  042704  000063              BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
2265  011146  005005                      CLR     R5                ;SET 'EXPECTED''
2266  011150  020504                      CMP     R5,R4             ;R5=GOOD; R4=?
2267  011152  001401                      BEQ     2$                ;BR IF OK
2268  011154  104003                      HLT     3                 ;COMPARISON ERROR
2269  011156  104400              2$:     SCOPE                     ;SCOPE THIS TEST
2270
2271
2272                                      ;************************ TEST 23 ******************************
2273                                      ;*LINE CONTROL REGISTER READ/WRITE TEST.
2274                                      ;*SET BIT10, VERIFY BIT10 WAS SET.
2275                                      ;*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.
2276                                      ;:***************************************************************
2277
2278                                  ;  TEST 23
2279                                  ;--------------
2280  011160  012737  000023  001226  TST23:  MOV     #23,TSTNO
2281  011166  012737  011244  001216          MOV     #TST24,NEXT
2282  011174  013703  001370                  MOV     DVLCR,R3          ;SET REGISTER TO BE TESTED.
2283  011200  012705  002000                  MOV     #BIT10,R5         ;SET 'EXPECTED ''.
2284  011204  010513                          MOV     R5,(R3)           ;WRITE THE REGISTER.
2285  011206  011304                          MOV     (R3),R4           ;READ THE REGISTER.
2286  011210  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
2287  011214  020504                          CMP     R5,R4             ;R5=GOOD; R4=UNKNOWN.
2288  011216  001401                          BEQ     1$                ;ARE THEY THE SAME?
2289  011220  104003                          HLT     3                 ;COMPARISON ERROR.
2290  011222  040513                  1$:     BIC     R5,(R3)           ;CLEAR BIT10
2291  011224  011304                          MOV     (R3),R4           ;READ THE REGISTER.
2292  011226  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
2293  011232  005005                          CLR     R5                ;SET 'EXPECTED''
2294  011234  020504                          CMP     R5,R4             ;R5=GOOD; R4=?
2295  011236  001401                          BEQ     2$                ;BR IF OK
2296  011240  104003                          HLT     3                 ;COMPARISON ERROR
2297  011242  104400                  2$:     SCOPE                     ;SCOPE THIS TEST
2298
2299
2300                                      ;************************ TEST 24 ******************************
2301                                      ;*LINE CONTROL REGISTER READ/WRITE TEST.
2302                                      ;*SET BIT11, VERIFY BIT11 WAS SET.
2303                                      ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED.
2304                                      ;:***************************************************************
2305
2306                                  ;  TEST 24
2307                                  ;--------------
2308  011244  012737  000024  001226  TST24:  MOV     #24,TSTNO
2309  011252  012737  011330  001216          MOV     #TST25,NEXT
2310  011260  013703  001370                  MOV     DVLCR,R3          ;SET REGISTER TO BE TESTED.
```

```
 2311  011264  012705  004000                  MOV     #BIT11,R5       ;SET 'EXPECTED ''.
 2312  011270  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
 2313  011272  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 2314  011274  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2315  011300  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 2316  011302  001401                          BEQ     1$              ;ARE THEY THE SAME?
 2317  011304  104003                          HLT     3               ;COMPARISON ERROR.
 2318  011306  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT11
 2319  011310  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 2320  011312  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2321  011316  005005                          CLR     R5              ;SET 'EXPECTED''
 2322  011320  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
 2323  011322  001401                          BEQ     2$              ;BR IF OK
 2324  011324  104003                          HLT     3               ;COMPARISON ERROR
 2325  011326  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
 2326
 2327
 2328                                          ;************************ TEST 25 *******************************
 2329                                          ;*LINE CONTROL REGISTER READ/WRITE TEST.
 2330                                          ;*SET BIT12, VERIFY BIT12 WAS SET.
 2331                                          ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED.
 2332                                          ;;*************************************************************************
 2333
 2334                                  ;  TEST 25
 2335                                  ;--------------
 2336  011330  012737  000025  001226  TST25:  MOV     #25,TSTNO
 2337  011336  012737  011414  001216          MOV     #TST26,NEXT
 2338  011344  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
 2339  011350  012705  010000                  MOV     #BIT12,R5       ;SET 'EXPECTED ''.
 2340  011354  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
 2341  011356  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 2342  011360  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2343  011364  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 2344  011366  001401                          BEQ     1$              ;ARE THEY THE SAME?
 2345  011370  104003                          HLT     3               ;COMPARISON ERROR.
 2346  011372  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT12
 2347  011374  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 2348  011376  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2349  011402  005005                          CLR     R5              ;SET 'EXPECTED''
 2350  011404  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
 2351  011406  001401                          BEQ     2$              ;BR IF OK
 2352  011410  104003                          HLT     3               ;COMPARISON ERROR
 2353  011412  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
 2354
 2355
 2356                                          ;************************ TEST 26 *******************************
 2357                                          ;*LINE CONTROL REGISTER READ/WRITE TEST.
 2358                                          ;*SET BIT13, VERIFY BIT13 WAS SET.
 2359                                          ;*CLEAR BIT13, VERIFY BIT13 WAS CLEARED.
 2360                                          ;;*************************************************************************
 2361
 2362                                  ;  TEST 26
 2363                                  ;--------------
 2364  011414  012737  000026  001226  TST26:  MOV     #26,TSTNO
 2365  011422  012737  011500  001216          MOV     #TST27,NEXT
 2366  011430  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
```

```
 2367  011434  012705  020000                    MOV     #BIT13,R5       ;SET 'EXPECTED ''.
 2368  011440  010513                            MOV     R5,(R3)         ;WRITE THE REGISTER.
 2369  011442  011304                            MOV     (R3),R4         ;READ THE REGISTER.
 2370  011444  042704  000063                    BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2371  011450  020504                            CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 2372  011452  001401                            BEQ     1$              ;ARE THEY THE SAME?
 2373  011454  104003                            HLT     3               ;COMPARISON ERROR.
 2374  011456  040513                    1$:     BIC     R5,(R3)         ;CLEAR BIT13
 2375  011460  011304                            MOV     (R3),R4         ;READ THE REGISTER.
 2376  011462  042704  000063                    BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2377  011466  005005                            CLR     R5              ;SET 'EXPECTED''
 2378  011470  020504                            CMP     R5,R4           ;R5=GOOD; R4=?
 2379  011472  001401                            BEQ     2$              ;BR IF OK
 2380  011474  104003                            HLT     3               ;COMPARISON ERROR
 2381  011476  104400                    2$:     SCOPE                   ;SCOPE THIS TEST
 2382
 2383
 2384                                            ;************************ TEST 27 ******************************
 2385                                            ;*LINE CONTROL REGISTER READ/WRITE TEST.
 2386                                            ;*SET BIT14, VERIFY BIT14 WAS SET.
 2387                                            ;*CLEAR BIT14, VERIFY BIT14 WAS CLEARED.
 2388                                            ;:**************************************************************
 2389
 2390                                    ;  TEST 27
 2391                                    ;--------------
 2392  011500  012737  000027  001226    TST27:  MOV     #27,TSTNO
 2393  011506  012737  011564  001216            MOV     #TST30,NEXT
 2394  011514  013703  001370                    MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
 2395  011520  012705  040000                    MOV     #BIT14,R5       ;SET 'EXPECTED ''.
 2396  011524  010513                            MOV     R5,(R3)         ;WRITE THE REGISTER.
 2397  011526  011304                            MOV     (R3),R4         ;READ THE REGISTER.
 2398  011530  042704  000063                    BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2399  011534  020504                            CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 2400  011536  001401                            BEQ     1$              ;ARE THEY THE SAME?
 2401  011540  104003                            HLT     3               ;COMPARISON ERROR.
 2402  011542  040513                    1$:     BIC     R5,(R3)         ;CLEAR BIT14
 2403  011544  011304                            MOV     (R3),R4         ;READ THE REGISTER.
 2404  011546  042704  000063                    BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
 2405  011552  005005                            CLR     R5              ;SET 'EXPECTED''
 2406  011554  020504                            CMP     R5,R4           ;R5=GOOD; R4=?
 2407  011556  001401                            BEQ     2$              ;BR IF OK
 2408  011560  104003                            HLT     3               ;COMPARISON ERROR
 2409  011562  104400                    2$:     SCOPE                   ;SCOPE THIS TEST
 2410
 2411
 2412                                            ;************************ TEST 30 ******************************
 2413                                            ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
 2414                                            ;*SET BIT0, VERIFY BIT0 WAS SET.
 2415                                            ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED.
 2416                                            ;:**************************************************************
 2417
 2418                                    ;  TEST 30
 2419                                    ;--------------
 2420  011564  012737  000030  001226    TST30:  MOV     #30,TSTNO
 2421  011572  012737  011640  001216            MOV     #TST31,NEXT
 2422  011600  013703  001372                    MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
```

```
2423  011604  012705  000001                  MOV     #BIT0,R5        ;SET 'EXPECTED ''.
2424  011610  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
2425  011612  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2426  011614  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2427  011616  001401                          BEQ     1$              ;ARE THEY THE SAME?
2428  011620  104003                          HLT     3               ;COMPARISON ERROR.
2429  011622  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT0
2430  011624  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2431  011626  005005                          CLR     R5              ;SET 'EXPECTED''
2432  011630  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
2433  011632  001401                          BEQ     2$              ;BR IF OK
2434  011634  104003                          HLT     3               ;COMPARISON ERROR
2435  011636  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
2436
2437
2438                                          ;************************ TEST 31 *******************************
2439                                          ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
2440                                          ;*SET BIT1, VERIFY BIT1 WAS SET.
2441                                          ;*CLEAR BIT1, VERIFY BIT1 WAS CLEARED.
2442                                          ;:*************************************************************************
2443
2444                                  ;  TEST 31
2445                                  ;--------------
2446  011640  012737  000031  001226  TST31:  MOV     #31,TSTNO
2447  011646  012737  011714  001216          MOV     #TST32,NEXT
2448  011654  013703  001372                  MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
2449  011660  012705  000002                  MOV     #BIT1,R5        ;SET 'EXPECTED ''.
2450  011664  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
2451  011666  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2452  011670  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2453  011672  001401                          BEQ     1$              ;ARE THEY THE SAME?
2454  011674  104003                          HLT     3               ;COMPARISON ERROR.
2455  011676  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT1
2456  011700  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2457  011702  005005                          CLR     R5              ;SET 'EXPECTED''
2458  011704  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
2459  011706  001401                          BEQ     2$              ;BR IF OK
2460  011710  104003                          HLT     3               ;COMPARISON ERROR
2461  011712  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
2462
2463
2464                                          ;************************ TEST 32 *******************************
2465                                          ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
2466                                          ;*SET BIT2, VERIFY BIT2 WAS SET.
2467                                          ;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.
2468                                          ;:*************************************************************************
2469
2470                                  ;  TEST 32
2471                                  ;--------------
2472  011714  012737  000032  001226  TST32:  MOV     #32,TSTNO
2473  011722  012737  011770  001216          MOV     #TST33,NEXT
2474  011730  013703  001372                  MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
2475  011734  012705  000004                  MOV     #BIT2,R5        ;SET 'EXPECTED ''.
2476  011740  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
2477  011742  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2478  011744  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
```

```
2479  011746  001401                        BEQ     1$              ;ARE THEY THE SAME?
2480  011750  104003                        HLT     3               ;COMPARISON ERROR.
2481  011752  040513                1$:     BIC     R5,(R3)         ;CLEAR BIT2
2482  011754  011304                        MOV     (R3),R4         ;READ THE REGISTER.
2483  011756  005005                        CLR     R5              ;SET 'EXPECTED''
2484  011760  020504                        CMP     R5,R4           ;R5=GOOD; R4=?
2485  011762  001401                        BEQ     2$              ;BR IF OK
2486  011764  104003                        HLT     3               ;COMPARISON ERROR
2487  011766  104400                2$:     SCOPE                   ;SCOPE THIS TEST
2488
2489
2490                                        ;************************ TEST 33 *******************************
2491                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
2492                                        ;*SET BIT3, VERIFY BIT3 WAS SET.
2493                                        ;*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
2494                                        ;:***************************************************************************
2495
2496                                ;  TEST 33
2497                                ;--------------
2498  011770  012737  000033  001226 TST33:  MOV     #33,TSTNO
2499  011776  012737  012044  001216         MOV     #TST34,NEXT
2500  012004  013703  001372                 MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
2501  012010  012705  000010                 MOV     #BIT3,R5        ;SET 'EXPECTED ''.
2502  012014  010513                         MOV     R5,(R3)         ;WRITE THE REGISTER.
2503  012016  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2504  012020  020504                         CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2505  012022  001401                         BEQ     1$              ;ARE THEY THE SAME?
2506  012024  104003                         HLT     3               ;COMPARISON ERROR.
2507  012026  040513                1$:      BIC     R5,(R3)         ;CLEAR BIT3
2508  012030  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2509  012032  005005                         CLR     R5              ;SET 'EXPECTED''
2510  012034  020504                         CMP     R5,R4           ;R5=GOOD; R4=?
2511  012036  001401                         BEQ     2$              ;BR IF OK
2512  012040  104003                         HLT     3               ;COMPARISON ERROR
2513  012042  104400                2$:      SCOPE                   ;SCOPE THIS TEST
2514
2515
2516                                        ;************************ TEST 34 *******************************
2517                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
2518                                        ;*SET BIT8, VERIFY BIT8 WAS SET.
2519                                        ;*CLEAR BIT8, VERIFY BIT8 WAS CLEARED.
2520                                        ;:***************************************************************************
2521
2522                                ;  TEST 34
2523                                ;--------------
2524  012044  012737  000034  001226 TST34:  MOV     #34,TSTNO
2525  012052  012737  012120  001216         MOV     #TST35,NEXT
2526  012060  013703  001372                 MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
2527  012064  012705  000400                 MOV     #BIT8,R5        ;SET 'EXPECTED ''.
2528  012070  010513                         MOV     R5,(R3)         ;WRITE THE REGISTER.
2529  012072  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2530  012074  020504                         CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2531  012076  001401                         BEQ     1$              ;ARE THEY THE SAME?
2532  012100  104003                         HLT     3               ;COMPARISON ERROR.
2533  012102  040513                1$:      BIC     R5,(R3)         ;CLEAR BIT8
2534  012104  011304                         MOV     (R3),R4         ;READ THE REGISTER.
```

```
2535  012106  005005                          CLR     R5              ;SET 'EXPECTED'
2536  012110  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
2537  012112  001401                          BEQ     2$              ;BR IF OK
2538  012114  104003                          HLT     3               ;COMPARISON ERROR
2539  012116  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
2540
2541
2542                                          ;************************ TEST 35 ******************************
2543                                          ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
2544                                          ;*SET BIT9, VERIFY BIT9 WAS SET.
2545                                          ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED.
2546                                          ;:**************************************************************
2547
2548                                  ;  TEST 35
2549                                  ;--------------
2550  012120  012737  000035  001226  TST35:  MOV     #35,TSTNO
2551  012126  012737  012174  001216          MOV     #TST36,NEXT
2552  012134  013703  001372                  MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
2553  012140  012705  001000                  MOV     #BIT9,R5        ;SET 'EXPECTED '.
2554  012144  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
2555  012146  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2556  012150  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2557  012152  001401                          BEQ     1$              ;ARE THEY THE SAME?
2558  012154  104003                          HLT     3               ;COMPARISON ERROR.
2559  012156  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT9
2560  012160  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2561  012162  005005                          CLR     R5              ;SET 'EXPECTED'
2562  012164  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
2563  012166  001401                          BEQ     2$              ;BR IF OK
2564  012170  104003                          HLT     3               ;COMPARISON ERROR
2565  012172  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
2566
2567
2568                                          ;************************ TEST 36 ******************************
2569                                          ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
2570                                          ;*SET BIT10, VERIFY BIT10 WAS SET.
2571                                          ;*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.
2572                                          ;:**************************************************************
2573
2574                                  ;  TEST 36
2575                                  ;--------------
2576  012174  012737  000036  001226  TST36:  MOV     #36,TSTNO
2577  012202  012737  012250  001216          MOV     #TST37,NEXT
2578  012210  013703  001372                  MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
2579  012214  012705  002000                  MOV     #BIT10,R5       ;SET 'EXPECTED '.
2580  012220  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
2581  012222  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2582  012224  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2583  012226  001401                          BEQ     1$              ;ARE THEY THE SAME?
2584  012230  104003                          HLT     3               ;COMPARISON ERROR.
2585  012232  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT10
2586  012234  011304                          MOV     (R3),R4         ;READ THE REGISTER.
2587  012236  005005                          CLR     R5              ;SET 'EXPECTED'
2588  012240  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
2589  012242  001401                          BEQ     2$              ;BR IF OK
2590  012244  104003                          HLT     3               ;COMPARISON ERROR
```

```
                                        2$:     SCOPE                   ;SCOPE THIS TEST

                                                ;************************ TEST 37 ******************************
                                                ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                                ;*SET BIT11, VERIFY BIT11 WAS SET.
                                                ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED.
                                                ;;**************************************************************

                                        ;  TEST 37
                                        ;--------------
012250  012737  000037  001226          TST37:  MOV     #37,TSTNO
012256  012737  012324  001216                  MOV     #TST40,NEXT
012264  013703  001372                          MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
012270  012705  004000                          MOV     #BIT11,R5       ;SET 'EXPECTED ''.
012274  010513                                  MOV     R5,(R3)         ;WRITE THE REGISTER.
012276  011304                                  MOV     (R3),R4         ;READ THE REGISTER.
012300  020504                                  CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
012302  001401                                  BEQ     1$              ;ARE THEY THE SAME?
012304  104003                                  HLT     3               ;COMPARISON ERROR.
012306  040513                          1$:     BIC     R5,(R3)         ;CLEAR BIT11
012310  011304                                  MOV     (R3),R4         ;READ THE REGISTER.
012312  005005                                  CLR     R5              ;SET 'EXPECTED''
012314  020504                                  CMP     R5,R4           ;R5=GOOD; R4=?
012316  001401                                  BEQ     2$              ;BR IF OK
012320  104003                                  HLT     3               ;COMPARISON ERROR
012322  104400                          2$:     SCOPE                   ;SCOPE THIS TEST

                                                ;************************ TEST 40 ******************************
                                                ;*SECONDARY REGISTER ACCESS REG. READ/WRITE TEST
                                                ;*SET EACH INDIVIDUAL BIT;  VERIFY EACH INDIVIDUAL BIT SET.
                                                ;*CLEAR EACH INDIVIDUAL BIT; VERIFY CLEAR.
                                                ;;**************************************************************


                                        ;  TEST 40
                                        ;--------------
012324  012737  000040  001226          TST40:  MOV     #40,TSTNO
012332  012737  012426  001216                  MOV     #TST41,NEXT
012340  013703  001376                          MOV     DVSRA,R3                ;SET CSR POINTER INTO R3
012344  012705  177777                          MOV     #-1,R5          ;SET EXPECTED TO ALL 1'S
012350  010513                                  MOV     R5,(R3)         ;WRITE REGISTER WITH EXPECTED DATA
012352  011304                                  MOV     (R3),R4         ;READ THE REGISTER
012354  020504                                  CMP     R5,R4           ;WAS READ EQUAL TO WRITTEN??
012356  001401                                  BEQ     1$              ;BR IF YES
012360  104003                                  HLT     3               ;PRIMARY REGISTER READ/WRITE TEST
012362  005005                          1$:     CLR     R5              ;SET DATA TO ZERO
012364  010513                                  MOV     R5,(R3)         ;WRITE REGISTER
012366  011304                                  MOV     (R3),R4         ;READ REGISTER
012370  020504                                  CMP     R5,R4           ;REGISTER WRITTEN OK?
012372  001401                                  BEQ     2$              ;BR IF YES
012374  104003                                  HLT     3               ;PRIMARY REGISTER DATA ERROR
012376  012705  000001                  2$:     MOV     #1,R5           ;SET FOR 'FLOATING 1''
012402  010513                          3$:     MOV     R5,(R3)         ;WRITE DATA
012404  011304                                  MOV     (R3),R4         ;READ DATA
```

```
2647  012406  020504                        CMP     R5,R4           ;DATA OK?
2648  012410  001401                        BEQ     4$              ;BR IF YES
2649  012412  104003                        HLT     3               ;DATA ERROR
2650  012414  000241                4$:     CLC                     ;ZERO CPU CARRY
2651  012416  006105                        ROL     R5              ;UPDATE DATA
2652  012420  001370                        BNE     3$              ;BR IF MORE TO GO
2653  012422  005013                        CLR     (R3)            ;LEAVE REG ALL 0'S
2654  012424  104400                        SCOPE                   ;SCOPE THIS TEST.
2655
2656
2657                                        ;************************ TEST 41 ******************************
2658                                        ;*SPECIAL FUNCT. REGISTER READ/WRITE TEST
2659                                        ;*SET EACH INDIVIDUAL BIT;  VERIFY EACH INDIVIDUAL BIT SET.
2660                                        ;*CLEAR EACH INDIVIDUAL BIT; VERIFY CLEAR.
2661                                        ;;**************************************************************************
2662
2663
2664                                ;  TEST 41
2665                                ;-------------
2666  012426  012737  000041  001226 TST41:  MOV     #41,TSTNO
2667  012434  012737  012536  001216         MOV     #TST42,NEXT
2668  012442  012777  000010  166712         MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
2669  012450  013703  001400                 MOV     DVSFR,R3                ;SET CSR POINTER INTO R3
2670  012454  012705  177777                 MOV     #-1,R5          ;SET EXPECTED TO ALL 1'S
2671  012460  010513                         MOV     R5,(R3)         ;WRITE REGISTER WITH EXPECTED DATA
2672  012462  011304                         MOV     (R3),R4         ;READ THE REGISTER
2673  012464  020504                         CMP     R5,R4           ;WAS READ EQUAL TO WRITTEN??
2674  012466  001401                         BEQ     1$              ;BR IF YES
2675  012470  104003                         HLT     3               ;PRIMARY REGISTER READ/WRITE TEST
2676  012472  005005                1$:      CLR     R5              ;SET DATA TO ZERO
2677  012474  010513                         MOV     R5,(R3)         ;WRITE REGISTER
2678  012476  011304                         MOV     (R3),R4         ;READ REGISTER
2679  012500  020504                         CMP     R5,R4           ;REGISTER WRITTEN OK?
2680  012502  001401                         BEQ     2$              ;BR IF YES
2681  012504  104003                         HLT     3               ;PRIMARY REGISTER DATA ERROR
2682  012506  012705  000001        2$:     MOV     #1,R5           ;SET FOR 'FLOATING 1''
2683  012512  010513                3$:     MOV     R5,(R3)         ;WRITE DATA
2684  012514  011304                         MOV     (R3),R4         ;READ DATA
2685  012516  020504                         CMP     R5,R4           ;DATA OK?
2686  012520  001401                         BEQ     4$              ;BR IF YES
2687  012522  104003                         HLT     3               ;DATA ERROR
2688  012524  000241                4$:     CLC                     ;ZERO CPU CARRY
2689  012526  006105                         ROL     R5              ;UPDATE DATA
2690  012530  001370                         BNE     3$              ;BR IF MORE TO GO
2691  012532  005013                         CLR     (R3)            ;LEAVE REG ALL 0'S
2692  012534  104400                         SCOPE                   ;SCOPE THIS TEST.
2693
2694
2695                                        ;************************ TEST 42 ******************************
2696
2697                                        ;*NPR STATUS REG. TEST
2698                                        ;*TEST THAT NPR STATUS REG. CANNOT BE WRITTEN
2699                                        ;*READ THE NPR STATUS REG. AND STORE THE DATA;
2700                                        ;*COMPLEMENT THE DATA AND  WRITE THE NPR STATUS REG.
2701                                        ;*VERIFING THAT THE NPR STATUS REG. DID NOT CHANGE.
2702                                        ;;**************************************************************************
```

```
2703
2704                                    ;  TEST 42
2705                                    ;--------------
2706  012536  012737  000042  001226    TST42:  MOV     #42,TSTNO
2707  012544  012737  012600  001216            MOV     #TST43,NEXT
2708  012552  013703  001402                    MOV     DVNSR,R3
2709                                                                ;GET THE REGISTER.
2710  012556  011305                            MOV     (R3),R5     ;READ THE REGISTER INTO R5
2711  012560  010504                            MOV     R5,R4       ;SAVE REG INTO R4
2712  012562  005104                            COM     R4          ;MAKE R4 OPPOSITE TO REGISTER
2713  012564  010413                            MOV     R4,(R3)     ;WRITE THE REGISTER WITH THE COMPLIMENT
2714  012566  011304                            MOV     (R3),R4     ;READ THE REGISTER.
2715  012570  020504                            CMP     R5,R4       ;IS THE REGISTER EQUAL TO ZERO.
2716  012572  001401                            BEQ     1$          ;BR IF OK.
2717  012574  104003                            HLT     3           ;REGISTER NOT ZERO.
2718  012576  104400                    1$:     SCOPE               ;SCOPE THIS TEST.
2719
2720
2721                                            ;********************** TEST 43 **************************
2722                                            ;*DV11 RESERVED REGISTER READ/WRITE TEST.
2723                                            ;*SET BIT0, VERIFY BIT0 WAS SET.
2724                                            ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED.
2725                                            ;;*********************************************************
2726
2727                                    ;  TEST 43
2728                                    ;--------------
2729  012600  012737  000043  001226    TST43:  MOV     #43,TSTNO
2730  012606  012737  012654  001216            MOV     #TST44,NEXT
2731  012614  013703  001404                    MOV     RESV16,R3   ;SET REGISTER TO BE TESTED.
2732  012620  012705  000001                    MOV     #BIT0,R5    ;SET 'EXPECTED ''.
2733  012624  010513                            MOV     R5,(R3)     ;WRITE THE REGISTER.
2734  012626  011304                            MOV     (R3),R4     ;READ THE REGISTER.
2735  012630  020504                            CMP     R5,R4       ;R5=GOOD; R4=UNKNOWN.
2736  012632  001401                            BEQ     1$          ;ARE THEY THE SAME?
2737  012634  104003                            HLT     3           ;COMPARISON ERROR.
2738  012636  040513                    1$:     BIC     R5,(R3)     ;CLEAR BIT0
2739  012640  011304                            MOV     (R3),R4     ;READ THE REGISTER.
2740  012642  005005                            CLR     R5          ;SET 'EXPECTED''
2741  012644  020504                            CMP     R5,R4       ;R5=GOOD; R4=?
2742  012646  001401                            BEQ     2$          ;BR IF OK
2743  012650  104003                            HLT     3           ;COMPARISON ERROR
2744  012652  104400                    2$:     SCOPE               ;SCOPE THIS TEST
2745
2746
2747                                            ;********************** TEST 44 **************************
2748                                            ;*DV11 RESERVED REGISTER READ/WRITE TEST.
2749                                            ;*SET BIT1, VERIFY BIT1 WAS SET.
2750                                            ;*CLEAR BIT1, VERIFY BIT1 WAS CLEARED.
2751                                            ;;*********************************************************
2752
2753                                    ;  TEST 44
2754                                    ;--------------
2755  012654  012737  000044  001226    TST44:  MOV     #44,TSTNO
2756  012662  012737  012730  001216            MOV     #TST45,NEXT
2757  012670  013703  001404                    MOV     RESV16,R3   ;SET REGISTER TO BE TESTED.
2758  012674  012705  000002                    MOV     #BIT1,R5    ;SET 'EXPECTED ''.
```

```
2759  012700  010513                        MOV     R5,(R3)         ;WRITE THE REGISTER.
2760  012702  011304                        MOV     (R3),R4         ;READ THE REGISTER.
2761  012704  020504                        CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2762  012706  001401                        BEQ     1$              ;ARE THEY THE SAME?
2763  012710  104003                        HLT     3               ;COMPARISON ERROR.
2764  012712  040513                1$:     BIC     R5,(R3)         ;CLEAR BIT1
2765  012714  011304                        MOV     (R3),R4         ;READ THE REGISTER.
2766  012716  005005                        CLR     R5              ;SET 'EXPECTED''
2767  012720  020504                        CMP     R5,R4           ;R5=GOOD; R4=?
2768  012722  001401                        BEQ     2$              ;BR IF OK
2769  012724  104003                        HLT     3               ;COMPARISON ERROR
2770  012726  104400                2$:     SCOPE                   ;SCOPE THIS TEST
2771
2772
2773                                        ;************************ TEST 45 *****************************
2774                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
2775                                        ;*SET BIT2, VERIFY BIT2 WAS SET.
2776                                        ;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.
2777                                        ;;*************************************************************************
2778
2779                                ;  TEST 45
2780                                ;--------------
2781  012730  012737  000045  001226 TST45:  MOV     #45,TSTNO
2782  012736  012737  013004  001216         MOV     #TST46,NEXT
2783  012744  013703  001404                 MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
2784  012750  012705  000004                 MOV     #BIT2,R5        ;SET 'EXPECTED ''.
2785  012754  010513                         MOV     R5,(R3)         ;WRITE THE REGISTER.
2786  012756  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2787  012760  020504                         CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2788  012762  001401                         BEQ     1$              ;ARE THEY THE SAME?
2789  012764  104003                         HLT     3               ;COMPARISON ERROR.
2790  012766  040513                1$:      BIC     R5,(R3)         ;CLEAR BIT2
2791  012770  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2792  012772  005005                         CLR     R5              ;SET 'EXPECTED''
2793  012774  020504                         CMP     R5,R4           ;R5=GOOD; R4=?
2794  012776  001401                         BEQ     2$              ;BR IF OK
2795  013000  104003                         HLT     3               ;COMPARISON ERROR
2796  013002  104400                2$:      SCOPE                   ;SCOPE THIS TEST
2797
2798
2799                                        ;************************ TEST 46 *****************************
2800                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
2801                                        ;*SET BIT3, VERIFY BIT3 WAS SET.
2802                                        ;*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
2803                                        ;;*************************************************************************
2804
2805                                ;  TEST 46
2806                                ;--------------
2807  013004  012737  000046  001226 TST46:  MOV     #46,TSTNO
2808  013012  012737  013060  001216         MOV     #TST47,NEXT
2809  013020  013703  001404                 MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
2810  013024  012705  000010                 MOV     #BIT3,R5        ;SET 'EXPECTED ''.
2811  013030  010513                         MOV     R5,(R3)         ;WRITE THE REGISTER.
2812  013032  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2813  013034  020504                         CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2814  013036  001401                         BEQ     1$              ;ARE THEY THE SAME?
```

```
2815  013040  104003                    HLT     3               ;COMPARISON ERROR.
2816  013042  040513              1$:   BIC     R5,(R3)         ;CLEAR BIT3
2817  013044  011304                    MOV     (R3),R4         ;READ THE REGISTER.
2818  013046  005005                    CLR     R5              ;SET 'EXPECTED''
2819  013050  020504                    CMP     R5,R4           ;R5=GOOD; R4=?
2820  013052  001401                    BEQ     2$              ;BR IF OK
2821  013054  104003                    HLT     3               ;COMPARISON ERROR
2822  013056  104400              2$:   SCOPE                   ;SCOPE THIS TEST
2823
2824
2825                                    ;************************ TEST 47 ******************************
2826                                    ;*DV11 RESERVED REGISTER READ/WRITE TEST.
2827                                    ;*SET BIT4, VERIFY BIT4 WAS SET.
2828                                    ;*CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
2829                                    ;:**************************************************************************
2830
2831                              ;  TEST 47
2832                              ;--------------
2833  013060  012737  000047  001226 TST47: MOV     #47,TSTNO
2834  013066  012737  013134  001216       MOV     #TST50,NEXT
2835  013074  013703  001404               MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
2836  013100  012705  000020               MOV     #BIT4,R5        ;SET 'EXPECTED ''.
2837  013104  010513                       MOV     R5,(R3)         ;WRITE THE REGISTER.
2838  013106  011304                       MOV     (R3),R4         ;READ THE REGISTER.
2839  013110  020504                       CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2840  013112  001401                       BEQ     1$              ;ARE THEY THE SAME?
2841  013114  104003                       HLT     3               ;COMPARISON ERROR.
2842  013116  040513                 1$:   BIC     R5,(R3)         ;CLEAR BIT4
2843  013120  011304                       MOV     (R3),R4         ;READ THE REGISTER.
2844  013122  005005                       CLR     R5              ;SET 'EXPECTED''
2845  013124  020504                       CMP     R5,R4           ;R5=GOOD; R4=?
2846  013126  001401                       BEQ     2$              ;BR IF OK
2847  013130  104003                       HLT     3               ;COMPARISON ERROR
2848  013132  104400                 2$:   SCOPE                   ;SCOPE THIS TEST
2849
2850
2851                                    ;************************ TEST 50 ******************************
2852                                    ;*DV11 RESERVED REGISTER READ/WRITE TEST.
2853                                    ;*SET BIT5, VERIFY BIT5 WAS SET.
2854                                    ;*CLEAR BIT5, VERIFY BIT5 WAS CLEARED.
2855                                    ;:**************************************************************************
2856
2857                              ;  TEST 50
2858                              ;--------------
2859  013134  012737  000050  001226 TST50: MOV     #50,TSTNO
2860  013142  012737  013210  001216       MOV     #TST51,NEXT
2861  013150  013703  001404               MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
2862  013154  012705  000040               MOV     #BIT5,R5        ;SET 'EXPECTED ''.
2863  013160  010513                       MOV     R5,(R3)         ;WRITE THE REGISTER.
2864  013162  011304                       MOV     (R3),R4         ;READ THE REGISTER.
2865  013164  020504                       CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2866  013166  001401                       BEQ     1$              ;ARE THEY THE SAME?
2867  013170  104003                       HLT     3               ;COMPARISON ERROR.
2868  013172  040513                 1$:   BIC     R5,(R3)         ;CLEAR BIT5
2869  013174  011304                       MOV     (R3),R4         ;READ THE REGISTER.
2870  013176  005005                       CLR     R5              ;SET 'EXPECTED''
```

```
2871  013200  020504                        CMP     R5,R4           ;R5=GOOD; R4=?
2872  013202  001401                        BEQ     2$              ;BR IF OK
2873  013204  104003                        HLT     3               ;COMPARISON ERROR
2874  013206  104400                2$:     SCOPE                   ;SCOPE THIS TEST
2875
2876
2877                                        ;************************ TEST 51 *****************************
2878                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
2879                                        ;*SET BIT6, VERIFY BIT6 WAS SET.
2880                                        ;*CLEAR BIT6, VERIFY BIT6 WAS CLEARED.
2881                                        ;:**************************************************************
2882
2883                                ;  TEST 51
2884                                ;--------------
2885  013210  012737  000051  001226 TST51:  MOV     #51,TSTNO
2886  013216  012737  013264  001216         MOV     #TST52,NEXT
2887  013224  013703  001404                 MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
2888  013230  012705  000100                 MOV     #BIT6,R5        ;SET 'EXPECTED ''.
2889  013234  010513                         MOV     R5,(R3)         ;WRITE THE REGISTER.
2890  013236  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2891  013240  020504                         CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2892  013242  001401                         BEQ     1$              ;ARE THEY THE SAME?
2893  013244  104003                         HLT     3               ;COMPARISON ERROR.
2894  013246  040513                1$:      BIC     R5,(R3)         ;CLEAR BIT6
2895  013250  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2896  013252  005005                         CLR     R5              ;SET 'EXPECTED''
2897  013254  020504                         CMP     R5,R4           ;R5=GOOD; R4=?
2898  013256  001401                         BEQ     2$              ;BR IF OK
2899  013260  104003                         HLT     3               ;COMPARISON ERROR
2900  013262  104400                2$:     SCOPE                   ;SCOPE THIS TEST
2901
2902
2903                                        ;************************ TEST 52 *****************************
2904                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
2905                                        ;*SET BIT7, VERIFY BIT7 WAS SET.
2906                                        ;*CLEAR BIT7, VERIFY BIT7 WAS CLEARED.
2907                                        ;:**************************************************************
2908
2909                                ;  TEST 52
2910                                ;--------------
2911  013264  012737  000052  001226 TST52:  MOV     #52,TSTNO
2912  013272  012737  013340  001216         MOV     #TST53,NEXT
2913  013300  013703  001404                 MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
2914  013304  012705  000200                 MOV     #BIT7,R5        ;SET 'EXPECTED ''.
2915  013310  010513                         MOV     R5,(R3)         ;WRITE THE REGISTER.
2916  013312  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2917  013314  020504                         CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2918  013316  001401                         BEQ     1$              ;ARE THEY THE SAME?
2919  013320  104003                         HLT     3               ;COMPARISON ERROR.
2920  013322  040513                1$:      BIC     R5,(R3)         ;CLEAR BIT7
2921  013324  011304                         MOV     (R3),R4         ;READ THE REGISTER.
2922  013326  005005                         CLR     R5              ;SET 'EXPECTED''
2923  013330  020504                         CMP     R5,R4           ;R5=GOOD; R4=?
2924  013332  001401                         BEQ     2$              ;BR IF OK
2925  013334  104003                         HLT     3               ;COMPARISON ERROR
2926  013336  104400                2$:     SCOPE                   ;SCOPE THIS TEST
```

```
                                        ;************************ TEST 53 ******************************
                                        ;*TEST OF THE BYTE OPERATIONS FOR THE DV11
                                        ;*SYSTEM CONTROL REG AND THE SECONDARY REG SEL.
                                        ;*THE TEST WILL CLEAR DVSCR AND THE WRITE (LOW BYTE)
                                        ;*BIT3; THEN VERIFY ONLY BIT3 IS SET; THEN THE
                                        ;*TEST WILL WRITE BIT 8(HIGH BYTE) AND VERFIY THAT
                                        ;*BIT8 AND BIT3 ARE SET.  THE EXACT PROCEEDURE
                                        ;*WILL BE USED ON THE DVSRS REGISTER.
                                        ;;*************************************************************

                                ;  TEST 53
                                ;--------------
013340  012737  000053  001226  TST53:  MOV     #53,TSTNO
013346  012737  013474  001216          MOV     #TST54,NEXT
013354  013703  001362                  MOV     DVSCR,R3        ;SET DVSCR POINTER
013360  005013                          CLR     (R3)            ;MAKE SURE IT =0
013362  012705  000010                  MOV     #BIT3,R5        ;LOAD EXPECTED RESULTS
013366  110513                          MOVB    R5,(R3)         ;"WRITE" BYTE (LOW) BIT3
013370  011304                          MOV     (R3),R4         ;READ (WORD) RESULT
013372  020504                          CMP     R5,R4           ;MAKE SURE ONLY BIT3 IS SET
013374  001401                          BEQ     1$              ;BR IF OK
013376  104003                          HLT     3               ;DVSCR WRONG
013400  012705  000410          1$:     MOV     #BIT8+BIT3,R5   ;SET EXPECTED DATA
013404  112763  000001  000001          MOVB    #BIT0,1(R3)     ;"WRITE" BYTE (HIGH) BIT0 [BIT8 OF WORD]
013412  011304                          MOV     (R3),R4         ;READ (WORD) RESULT
013414  020504                          CMP     R5,R4           ;OK?
013416  001401                          BEQ     2$              ;
013420  104003                          HLT     3               ;DVSCR WRONG
013422  005013                  2$:     CLR     (R3)            ;LEAVE REGISTERED CLEARED
013424  013703  001372                  MOV     DVSRS,R3        ;GET NEXT REGISTER FOR BYTE TEST.
013430  005013                          CLR     (R3)            ;MAKE SURE WORD IS =0.
013432  012705  000010                  MOV     #BIT3,R5        ;SET EXPECTED
013436  110513                          MOVB    R5,(R3)         ;WRITE BYTE (LOW) BIT3
013440  011304                          MOV     (R3),R4         ;READ RESULT
013442  020504                          CMP     R5,R4           ;OK?
013444  001401                          BEQ     4$              ;
013446  104003                          HLT     3               ;DVSRS WRONG
013450  012705  000410          4$:     MOV     #BIT8+BIT3,R5   ;SET EXPECTED
013454  112763  000001  000001          MOVB    #BIT0,1(R3)     ;WRITE BYTE (HIGH) BIT0 [BIT8 OF WORD]
013462  011304                          MOV     (R3),R4         ;READ RESULT
013464  020504                          CMP     R5,R4           ;OK
013466  001401                          BEQ     5$              ;
013470  104003                          HLT     3               ;DVSRS FAILED
013472  104400                  5$:     SCOPE                   ;SCOPE TEST
```

```
                                        ;************************ TEST 54 ******************************
                                        ;*SECONDARY REGISTER READ/WRITE TESTS
                                        ;*READ/WRITE TEST. READ AND WRITE DIFFERENT DATA
                                        ;*PATTERENS INTO THE SECONDARY REGISTERS VERIFING
                                        ;*THAT WHAT WAS READ MATCHES WHAT WAS WRITTEN.
                                        ;:**************************************************************

                                ;  TEST 54
                                ;--------------
013474  012737  000054  001226  TST54:  MOV     #54,TSTNO
013502  012737  013676  001216          MOV     #TST55,NEXT
013510  012737  013526  001220          MOV     #1$,LOCK
013516  005000                          CLR     R0              ;SEL LINE NUMBER TO ZERO
013520  005001                          CLR     R1              ;SET SEC. REG TO ZERO ALSO
013522  013702  001376                  MOV     DVSRA,R2        ;SET ADDRESS POINTER INTO R2
013526  110077  165640          1$:     MOVB    R0,@DVSRS       ;LOAD LINE NUMBER
013532  110177  165636                  MOVB    R1,@DVSRSH      ;LOAD SEC. REG.
013536  005004                          CLR     R4              ;ZERO DATA PATTERN
013540  010412                  2$:     MOV     R4,(R2)         ;LOAD DATA INTO DV11 REGISTER
013542  011203                          MOV     (R2),R3         ;READ BACK THE DATA
013544  020403                          CMP     R4,R3           ;WAS WHAT WAS WRITTEN READ BACK??
013546  001401                          BEQ     64$             ;BR IF DATA OK
013550  104002                          HLT     2               ;SECONDARY REGISTER READ WRITE ERROR.
013552  104401                  64$:    SCOP1                   ;LOCK ON DATA,LINE,AND SEC REG? (SW09=1)
013554  005104                          COM     R4              ;MAKE DATA ALL 1'S
013556  001370                          BNE     2$              ;RETURN AND WRITE IT INTO THE REGISTER
013560  012704  000001                  MOV     #1,R4           ;GET READY FOR THE ''FLOATING 1''
013564  012737  013572  001220          MOV     #3$,LOCK        ;SET IF SW09=1
013572  010412                  3$:     MOV     R4,(R2)         ;LOAD DATA
013574  011203                          MOV     (R2),R3         ;READ DATA
013576  020403                          CMP     R4,R3           ;DATA OK??
013600  001401                          BEQ     65$             ;BR IF OK!
013602  104002                          HLT     2               ;SECONDARY REGISTER READ/WRITE ERROR
013604  104401                  65$:    SCOP1                   ;SW09=1?
013606  000241                          CLC                     ;ZERO CPU CARRY
013610  006104                          ROL     R4              ;FLOAT THE 1
013612  001367                          BNE     3$              ;IF NOT DONE WRITE NEW PATTERN
013614  012704  177776                  MOV     #^C<1>,R4       ;GET READY FOR THE ''FLOATING 0''
013620  012737  013626  001220          MOV     #4$,LOCK        ;SET IF SW09=1
013626  010412                  4$:     MOV     R4,(R2)         ;WRITE DATA
013630  011203                          MOV     (R2),R3         ;READ DATA
013632  020403                          CMP     R4,R3           ;DATA OK??
013634  001401                          BEQ     66$             ;BR IF OK!
013636  104002                          HLT     2               ;SECONDARY REGISTER DATA COMPARE ERROR
013640  104401                  66$:    SCOP1                   ;SW09=1??
013642  000261                          SEC                     ;SET CPU CARRY
013644  006104                          ROL     R4              ;CHANGE DATA PATTERN
013646  103767                          BCS     4$              ;IF NOT DONE ,GO BACK WITH NEW PATTERN
013650  005012                          CLR     (R2)            ;ZERO THE REGISTER (ALL DONE WITH THIS ONE.
013652  005201                          INC     R1              ;UPDATE THE SEC REG POINTER
013654  022701  000020                  CMP     #16.,R1         ;ALL SEC REGISTERS DONE?
013660  001322                          BNE     1$              ; BR IF NO.
013662  005001                          CLR     R1              ;ZERO SEC REG POINTER
013664  005200                          INC     R0              ;UPDATE LINE NUMBER POINTER
013666  022700  000020                  CMP     #16.,R0         ;ALL LINES DONE?
013672  001315                          BNE     1$              ;BR IF NO
```

```
3030  013674  104400                          SCOPE                   ;SCOPE THIS TEST
3031
3032
3033                                          ;************************ TEST 55 ******************************
3034                                          ;*INDIVIDUAL LINE DUEL ADDRESS TESTS
3035                                          ;*THIS TEST VERIFIES THAT WRITING ONE SECONDARY
3036                                          ;*REGISTER FOR A SPECIFIC LINE DOES NOT ALTER
3037                                          ;*ANY OTHER SECONDARY REGISTER FOR THAT LINE.
3038                                          ;;**************************************************************************
3039
3040                                  ;  TEST 55
3041                                  ;--------------
3042  013676  012737  000055  001226  TST55:  MOV     #55,TSTNO
3043  013704  012737  014036  001216          MOV     #TST56,NEXT
3044  013712  012737  013772  001220          MOV     #2$,LOCK
3045  013720  005000                          CLR     R0              ;SELECT THE LINE NUMBER.
3046  013722  005001                  65$:    CLR     R1              ;SET FOR SEC. REG. POINTER.
3047  013724  005004                          CLR     R4              ;SET DATA
3048  013726  013702  001376                  MOV     DVSRA,R2        ;SET ACCESS REGISTER.
3049  013732  110077  165434                  MOVB    R0,@DVSRS       ;SELECT THE LINE NUMBER
3050  013736  110177  165432          1$:     MOVB    R1,@DVSRSH      ;SELECT THE SEC. REG.
3051  013742  010412                          MOV     R4,(R2)         ;WRITE SEC. REG.
3052  013744  062704  010421                  ADD     #^B<0001000100010001>,R4
3053                                                                  ;UPDATE DATA
3054  013750  005201                          INC     R1              ;UPDATE SECONDARY REG. POINTER
3055  013752  022701  000020                  CMP     #16.,R1         ;ALL SEC. REG. DONE?
3056  013756  001367                          BNE     1$              ;BR IF NO
3057  013760  005001                          CLR     R1              ;RESET SEC. REG. POINTER TO ZERO.
3058  013762  005004                          CLR     R4              ;ZERO DATA COMPARE
3059  013764  012737  013772  001220          MOV     #2$,LOCK        ;SET FOR LOCK.
3060  013772  110177  165376          2$:     MOVB    R1,@DVSRSH      ;GET SEC. REG.
3061  013776  011203                          MOV     (R2),R3         ;READ SEC. REG.
3062  014000  020403                          CMP     R4,R3           ;R4=GOOD; R3=UNKNOWN
3063  014002  001401                          BEQ     3$              ;BR IF ALL OK
3064  014004  104002                          HLT     2               ;SECONDARY REGISTER ADDRESSING ERROR
3065  014006  104401                  3$:     SCOP1                   ;LOCK ON REG. (SW09=1)
3066  014010  062704  010421                  ADD     #^B<0001000100010001>,R4
3067  014014  005201                          INC     R1              ;UPDATE SEC REG POINTER
3068  014016  022701  000020                  CMP     #16.,R1         ;ALL 16 LINES TESTED YET?
3069  014022  001363                          BNE     2$              ;BR IF NO
3070  014024  005200                          INC     R0              ;UPDATE LINE NO POINTER
3071  014026  022700  000020                  CMP     #16.,R0         ;ALL LINES DONE??
3072  014032  001333                          BNE     65$             ;BR IF NO
3073  014034  104400                          SCOPE                   ;SCOPE THE TEST
3074
3075
3076                                          ;************************ TEST 56 ******************************
3077                                          ;*VERIFY NO LINE INTERACTION.
3078                                          ;*THIS TEST VERIFIES THAT WRITING THE SECONDARY
3079                                          ;*REGISTERS FOR ONE LINE DOES NOT INTERFEAR WITH
3080                                          ;*THE SECONDARY REGISTERS OF ANOTHER LINE.
3081                                          ;;**************************************************************************
3082
3083                                  ;  TEST 56
3084                                  ;--------------
3085  014036  012737  000056  001226  TST56:  MOV     #56,TSTNO
```

```
3086  014044  012737  000002  001222          MOV     #2,ICOUNT
3087  014052  012737  014434  001216          MOV     #TST57,NEXT
3088  014060  012737  014152  001220          MOV     #4$,LOCK
3089  014066  005000                          CLR     R0              ;SELECT THE LINE NUMBER
3090  014070  005001                          CLR     R1              ;R1 = SECONDARY REGISTER SELECT.
3091  014072  005004                          CLR     R4              ;SET DATA TO BE WRITTEN.
3092  014074  013702  001376                  MOV     DVSRA,R2        ;SET ''SECONDARY REGISTER ACCESS'' POINTER.
3093  014100  110077  165266          1$:     MOVB    R0,@DVSRS       ;GET THE LINE NUMBER SELECTED.
3094  014104  110177  165264          2$:     MOVB    R1,@DVSRSH      ;GET THE SECONDARY REGISTER.
3095  014110  010412                          MOV     R4,(R2)         ;WRITE THE SECONDARY REG.
3096  014112  005201                          INC     R1              ;UPDATE  SECONDARY REG. POINTER.
3097  014114  022701  000020                  CMP     #16.,R1         ;ALL SEC. REG. DONE?
3098  014120  001371                          BNE     2$              ;BR IF NO.
3099  014122  005001                          CLR     R1              ;ZERO SEC.REG.POINTER
3100  014124  062704  010421                  ADD     #^B<0001000100010001>,R4
3101                                                                  ;UPDATE DATA POINTER.
3102  014130  005200                          INC     R0              ;UPDATE LINE POINTER.
3103  014132  022700  000020                  CMP     #16.,R0         ;ALL LINES DONE?
3104  014136  001360                          BNE     1$              ;BR IF NO.
3105
3106  014140  005000                          CLR     R0              ;SELECT LINE NUMBER
3107  014142  005001                          CLR     R1              ;START SEC. REG. AT 0
3108  014144  005004                          CLR     R4              ;ZERO DATA COMPARE.
3109  014146  110077  165220          3$:     MOVB    R0,@DVSRS       ;SELECT LINE NUMBER.
3110  014152  110177  165216          4$:     MOVB    R1,@DVSRSH      ;SELECT SEC.REG.
3111  014156  011203                          MOV     (R2),R3         ;READ THE SEC. REG.
3112  014160  020403                          CMP     R4,R3           ;WAS DATA CORRECT?
3113  014162  001401                          BEQ     5$              ;BR IF OK
3114  014164  104002                          HLT     2               ;SECONDARY REG. ADDRESSING ERROR.
3115  014166  104401                  5$:     SCOP1                   ;LOCK ON REGISTER? (SW09=1)
3116  014170  005201                          INC     R1              ;UPDATE SEC. REG. POINTER
3117  014172  022701  000020                  CMP     #16.,R1         ;ALL SEC. REG. FOR THIS LINE DONE?
3118  014176  001365                          BNE     4$              ;BR IF NO.
3119  014200  062704  010421                  ADD     #^B<0001000100010001>,R4
3120                                                                  ;UPDATE DATA
3121  014204  005001                          CLR     R1              ;SET FOR SEC. REG.
3122  014206  005200                          INC     R0              ;UPDATE LINE NUMBER POINTER.
3123  014210  022700  000020                  CMP     #16.,R0         ;ALL LINES DONE?
3124  014214  001354                          BNE     3$              ;BR IF NO
3125
3126                                          ;*PART 2
3127                                          ;*FILL ALL RAMS WITH ALL 1'S AND THEN
3128                                          ;*CLEAR JUST ONE BIT AT A TIME VERIFYING
3129                                          ;*THAT ONLY THAT ONE BIT IS CLEAR AND THAT
3130                                          ;*ALL OTHER RAMS STILL CONTAIN ALL 1'S.
3131                                          ;*THERE SHOULD BE ONLY ONE BIT CLEARED AT
3132                                          ;*ONE TIME.
3133
3134  014216  005077  165162                  CLR     @RESV16         ;CLEAR ''LOC FOUND FLAG''.
3135  014222  005037  001220                  CLR     LOCK            ;
3136  014226  013702  001376          MAR17:  MOV     DVSRA,R2        ;LOAD POINTER TO DVSRA
3137  014232  005077  165134                  CLR     @DVSRS          ;SET LINE AND SEC REG POINTER TO ZERO
3138  014236  012712  177777          1$:     MOV     #^B<1111111111111111>,(R2)
3139                                                                  ;PREPARE TO LOAD ALL 1'S INTO ALL RAM LOC.
3140  014242  062777  170361  165122          ADD     #^C<BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+BIT0,@DVSRS
3141                                                                  ;UPDATE LINE AND SEC REG POINTERS.
```

```
3142  014250  001372                        BNE     1$              ;BR IF NOT ALL DONE FILLING 1'S.
3143  014252  005000                        CLR     R0              ;ZERO LINE # POINTER
3144  014254  005001                        CLR     R1              ;ZERO  SEC REG # POINTER
3145  014256  012704  177776        2$:     MOV     #^C<BIT0>,R4    ;SET INITAL DATA
3146  014262  110077  165104        3$:     MOVB    R0,@DVSRS       ;LOAD LINE #
3147  014266  110177  165102                MOVB    R1,@DVSRSH      ;LOAD SEC REG #
3148  014272  010412                        MOV     R4,(R2)         ;LOAD DATA
3149  014274  005077  165072                CLR     @DVSRS          ;ZERO POINTERS
3150  014300  022712  177777        100$:   CMP     #^B<1111111111111111>,(R2)
3151                                                                ;VERIFY ONLY ONE LOC. HAS ONE BIT CLEARED
3152  014304  001417                        BEQ     6$              ;BR IF LOC. OK
3153  014306  012777  177777  165070        MOV     #-1,@RESV16     ;SET 'LOC FOUND FLAG'.
3154  014314  011203                        MOV     (R2),R3         ;SAVE DATA
3155  014316  120077  165050                CMPB    R0,@DVSRS       ;IS THIS THE RIGHT LINE?
3156  014322  001401                        BEQ     4$              ;BR IF YES
3157  014324  104002                        HLT     2               ;WRONG LINE HAS CLEARED BIT!
3158  014326  120177  165042        4$:     CMPB    R1,@DVSRSH      ;IS THIS THE RIGHT SEC REG?
3159  014332  001401                        BEQ     5$              ;BR IF YES
3160  014334  104002                        HLT     2               ;WRONG SEC REG HAS CLEARED BIT.
3161  014336  020403                5$:     CMP     R4,R3           ;IS THE ACTUAL DATA OK?
3162  014340  001401                        BEQ     6$              ;BR IF YES
3163  014342  104002                        HLT     2               ;ACTUAL DATA WAS WRONG.
3164  014344  062777  170361  165020  6$:     ADD     #^C<BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+BIT0,@DVSRS
3165  014352  001352                        BNE     100$            ;BR IF NOT DONE.
3166  014354  005777  165024                TST     @RESV16         ;HAS A LOC BEEN FOUND?
3167  014360  001001                        BNE     7$              ;BR IF YES
3168  014362  104000                        HLT     0               ;NO LOC WAS FOUND WITH A ZERO BIT.
3169  014364  005077  165014        7$:     CLR     @RESV16         ;CLEAR 'LOC FOUND FLAG'.
3170  014370  000261                        SEC                     ;SHIFT IN A  1
3171  014372  006104                        ROL     R4              ;CHANGE DATA PATERN
3172  014374  103732                        BCS     3$              ;DO IT ALL OVER AGAIN
3173  014376  110077  164770                MOVB    R0,@DVSRS       ;LOAD LINE NO.
3174  014402  110177  164766                MOVB    R1,@DVSRSH      ;LOAD SEC REG.
3175  014406  010412                        MOV     R4,(R2)         ;PUT RAM LOC BACK TO ALL 1'S.
3176  014410  005201                        INC     R1              ;UPDATE SEC REG #
3177  014412  022701  000020                CMP     #16.,R1         ;ALL SEC REG DONE?
3178  014416  001317                        BNE     2$              ;BR IF NO
3179  014420  005001                        CLR     R1              ;ZERO SEC REG POINTER
3180  014422  005200                        INC     R0              ;UPDATE  LINE POINTER
3181  014424  022700  000020                CMP     #16.,R0         ;ALL LINES DONE?
3182  014430  001312                        BNE     2$              ;BR IF NO
3183  014432  104400                        SCOPE                   ;SCOPE  THIS TEST.
3184
3185
3186                                ;************************** TEST 57 *********************************
3187                                ;*MEMORY EXTENSION READ/WRITE TEST
3188                                ;*VERIFY BITS 4 AND 5 OF EACH LINE
3189                                ;*SECONDARY REGISTERS EXERCISED ARE:
3190                                ;*  00    TX BUS ADDRESS (PRIMARY)
3191                                ;*  02    TX BUS ADDRESS (SECONDARY)
3192                                ;*  04    RX BUS ADDRESS
3193                                ;*  10    TX TABLE BASE ADDRESS
3194                                ;*  11    RX TABLE BASE ADDRESS
3195                                ;*NOTE THAT ALL LINES (00-16) ARE EXERCISED.
3196                                ;:*********************************************************************
3197
```

```
3198
3199                                    ;  TEST 57
3200                                    ;--------------
3201  014434  012737  000057  001226    TST57:  MOV     #57,TSTNO
3202  014442  012737  014630  001216            MOV     #TST60,NEXT
3203  014450  012737  014510  001220            MOV     #2$,LOCK
3204  014456  005000                            CLR     R0              ;R0=LINE NUMBER (START AT 0)
3205  014460  013702  001370                    MOV     DVLCR,R2        ;SET R2 =BASE ADDRESS(DVLCR)
3206  014464  012704  000060            1$:     MOV     #BIT5+BIT4,R4   ;R4=GOOD DATA  (BOTH EA BITS SET AT START)
3207  014470  012705  000005                    MOV     #5,R5           ;R5 IS COUNTER OF SEC REGISTERS
3208  014474  012737  000004  001246            MOV     #4,TEMP1        ;TEMP1 IS COUNTER OF COMB. OF EA BITS.
3209                                                                    ;EX: 11,10,01,00
3210  014502  012737  014622  001256            MOV     #MEMEXT,TEMP5   ;TEMP5=SEC. REGISTER POINTER.
3211  014510  010477  164646            2$:     MOV     R4,@DVSCR       ;LOAD DVSCR WITH EA BITS.
3212  014514  110077  164652                    MOVB    R0,@DVSRS       ;SEL THE LINE NUMBER
3213  014520  117701  164532                    MOVB    @TEMP5,R1       ;GET SEC REG.
3214  014524  110177  164644                    MOVB    R1,@DVSRSH      ;SEL THE SEC. REGISTER
3215  014530  005077  164642                    CLR     @DVSRA          ;HIT THE SEC.REG. ACCESS REGISTER.
3216  014534  017703  164630                    MOV     @DVLCR,R3       ;SAVE THE DVLINE PARM. REG.
3217  014540  042703  177717                    BIC     #^C<BIT5+BIT4>,R3
3218                                                                    ;CLEAR ALL BUT BITS 5 AND 4.
3219  014544  020403                            CMP     R4,R3           ;ARE THE EA BITS GOOD
3220  014546  001401                            BEQ     3$
3221  014550  104004                            HLT     4
3222  014552  104401                    3$:     SCOP1                   ;SW09=1?
3223  014554  005237  001256                    INC     TEMP5           ;POP POINTER
3224  014560  005305                            DEC     R5              ;ALL SEC REG DONE?
3225  014562  001352                            BNE     2$              ;BR IF NO.
3226  014564  012737  014622  001256            MOV     #MEMEXT,TEMP5   ;RESET POINTER
3227  014572  012705  000005                    MOV     #5,R5           ;RESET COUNTER
3228  014576  162704  000020                    SUB     #BIT4,R4        ;ADJUST FOR NEXT EA BIT PATTERN
3229  014602  005337  001246                    DEC     TEMP1           ;ALL PATERNS DONE?
3230  014606  001340                            BNE     2$
3231  014610  005200                            INC     R0              ;UPDATE TO NEXT LINE
3232  014612  022700  000020                    CMP     #16.,R0         ;ALL LINES DONE
3233  014616  001322                            BNE     1$              ;BR IF NO.
3234  014620  104400                            SCOPE
3235                                    ;TABLE OF SECONDARY REGISTERS EXERCISERD....
3236  014622      000                   MEMEXT: .BYTE   00
3237  014623      002                           .BYTE   02
3238  014624      004                           .BYTE   04
3239  014625      010                           .BYTE   10
3240  014626      011                           .BYTE   11
3241        014630                      .EVEN
3242
3243
3244                                            ;************************* TEST 60 ******************************
3245                                            ;*INITIALIZATION TESTS
3246                                            ;*SET ALL POSSIBLE BITS IN ALL THE PRIMARY REGISTERS
3247                                            ;*AND VERIFY THAT ALL THE BITS ARE CLEARED
3248                                            ;*BY A BUS RESET
3249                                            ;;*************************************************************************
3250
3251                                    ;  TEST 60
3252                                    ;--------------
3253  014630  012737  000060  001226    TST60:  MOV     #60,TSTNO
```

```
3254  014636  012737  015044  001216          MOV     #TST61,NEXT
3255  014644  012737  000340  177776          MOV     #340,PS         ;LOCK OUT INTERUPTS.
3256  014652  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER POINTER FOR LOADING
3257  014656  005077  164510                  CLR     @DVSRS          ;CLEAR LINE POINTER
3258  014662  005077  164510                  CLR     @DVSRA          ;CLEAR ACCESS REG.
3259  014666  012723  173777                  MOV     #^C<BIT11>,(R3)+
3260  014672  012702  000007                  MOV     #7,R2           ;SET ALL BITS BUT MSTCLR
3261  014676  012723  177777          1$:     MOV     #-1,(R3)+       ;LOAD ALL OTHER REGISTERS WITH ALL 1'S
3262  014702  005302                          DEC     R2              ;ALLREGISTERS LOADED?
3263  014704  001374                          BNE     1$              ;BR IF NO
3264  014706  000005                          RESET                   ;ISSUES A BUS INIT (RESET INSTR)
3265  014710  005200                          INC     R0              ;FLASH THE CPU LIGHTS!!!
3266  014712  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER POINTER
3267  014716  005005                          CLR     R5              ;SET 'EXPECTED" FOR DVSCR
3268  014720  011304                          MOV     (R3),R4         ;READ THE DVSCR REG.
3269  014722  020504                          CMP     R5,R4           ;IS BIT8 ALONE SET?
3270  014724  001401                          BEQ     2$              ;BR IF YES
3271  014726  104003                          HLT     3               ;DVSCR HAS WRONG DATA>
3272  014730  005723                  2$:     TST     (R3)+           ;POP POINTER TO DVRIC
3273  014732  005005                          CLR     R5              ;SET EXPECTED TO ZERO
3274  014734  011304                          MOV     (R3),R4         ;DVRIC (EXPECT ALL 0'S)
3275  014736  001401                          BEQ     3$              ;BR IF OK
3276  014740  104003                          HLT     3               ;DVRIC NO ALL 0'S
3277  014742  005723                  3$:     TST     (R3)+           ;POP POINTER TO DVLCR REG
3278  014744  011304                          MOV     (R3),R4         ;DVLCR (READ DVLCR INTO R4)
3279  014746  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4
3280  014752  020504                          CMP     R5,R4           ;DISREGUARD BR TEST POINTS AND MEM EXT BITS.
3281  014754  001401                          BEQ     4$              ;DVLCR OK?
3282  014756  104003                          HLT     3               ;DVLCR INCORECT (DISREGUARD BIT5,4,1,0)
3283  014760  005723                  4$:     TST     (R3)+           ;POP POINTER TO DVSRS REG
3284  014762  011304                          MOV     (R3),R4         ;DVSRS (EXPECT ALL 0'S)
3285  014764  001401                          BEQ     5$              ;BR IF OK
3286  014766  104003                          HLT     3               ;DVSRS REG NOT ALL ZEROS
3287  014770  005723                  5$:     TST     (R3)+           ;POP POINTER TO DVSRA REG
3288  014772  011304                          MOV     (R3),R4         ;DVSRA (EXPECT ALL 0'S)
3289  014774  001401                          BEQ     6$              ;BR IF GOOD
3290  014776  104003                          HLT     3               ;DVSRA NOT ALL 0'S
3291  015000  005723                  6$:     TST     (R3)+           ;POP POINTER TO DVSFR
3292  015002  011304                          MOV     (R3),R4         ;DVSFR (EXPECT ALL 1'S (THATS RIGHT))
3293  015004  012705  177777                  MOV     #177777,R5      ;SET EXPECTED
3294  015010  020504                          CMP     R5,R4           ;EXPECETD =FOUND?
3295  015012  001401                          BEQ     7$              ;BR IF YES
3296  015014  104003                          HLT     3               ;DVSFR NOT ALL 1'S
3297  015016  005723                  7$:     TST     (R3)+           ;POP POINTER TO DVNSR REG
3298  015020  005713                          TST     (R3)            ;DVNSR S/B PLUS (15=0)
3299  015022  100001                          BPL     64$
3300  015024  104000                          HLT     0
3301  015026  005723                  64$:    TST     (R3)+           ;POP POINTER TO RESV16 REG
3302  015030  011304                          MOV     (R3),R4         ;RESV16 (EXPECT ALL 0'S)
3303  015032  005005                          CLR     R5              ;SET EXPECTED TO 0'S
3304  015034  020504                          CMP     R5,R4           ;WELL DOES IT =1'S?
3305  015036  001401                          BEQ     8$              ;BR IF OK
3306  015040  104003                          HLT     3               ;RESV16 NOT ALL 0'S
3307  015042  104400                  8$:     SCOPE                   ;SCOPE THIS TEST;
3308
3309
```

```
                                        ;************************ TEST 61 ******************************
                                        ;*INITIALIZATION TESTS
                                        ;*SET ALL POSSIBLE BITS IN ALL THE PRIMARY REGISTERS
                                        ;*AND VERIFY THAT ALL THE BITS ARE CLEARED
                                        ;*BY A MASTER CLEAR
                                        ;:*************************************************************************
                                        ;
                                ;  TEST 61
                                ;--------------
                                ;
015044  012737  000061  001226  TST61:  MOV     #61,TSTNO
015052  012737  015262  001216          MOV     #TST62,NEXT
015060  012737  000340  177776          MOV     #340,PS         ;LOCK OUT INTERUPTS.
015066  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER POINTER FOR LOADING
015072  005077  164274                  CLR     @DVSRS          ;CLEAR LINE POINTER
015076  005077  164274                  CLR     @DVSRA          ;CLEAR ACCESS REG.
015102  012723  173777                  MOV     #^C<BIT11>,(R3)+
015106  012702  000007                  MOV     #7,R2           ;SET ALL BITS BUT MSTCLR
015112  012723  177777          1$:     MOV     #-1,(R3)+       ;LOAD ALL OTHER REGISTERS WITH ALL 1'S
015116  005302                  DEC     R2              ;ALLREGISTERS LOADED?
015120  001374                  BNE     1$              ;BR IF NO
015122  052777  004000  164232          BIS     #MRESET,@DVSCR  ;ISSUE A 'MASTER CLEAR''
015130  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER POINTER
015134  005005                  CLR     R5              ;SET 'EXPECTED'' FOR DVSCR
015136  011304                  MOV     (R3),R4         ;READ THE DVSCR REG.
015140  020504                  CMP     R5,R4           ;IS BIT8 ALONE SET?
015142  001401                  BEQ     2$              ;BR IF YES
015144  104003                  HLT     3               ;DVSCR HAS WRONG DATA>
015146  005723          2$:     TST     (R3)+           ;POP POINTER TO DVRIC
015150  005005                  CLR     R5              ;SET EXPECTED TO ZERO
015152  011304                  MOV     (R3),R4         ;DVRIC (EXPECT ALL 0'S)
015154  001401                  BEQ     3$              ;BR IF OK
015156  104003                  HLT     3               ;DVRIC NO ALL 0'S
015160  005723          3$:     TST     (R3)+           ;POP POINTER TO DVLCR REG
015162  011304                  MOV     (R3),R4         ;DVLCR (READ DVLCR INTO R4)
015164  042704  000063          BIC     #BIT5+BIT4+BIT1+BIT0,R4
015170  020504                  CMP     R5,R4           ;DISREGUARD BR TEST POINTS AND MEM EXT BITS.
015172  001401                  BEQ     4$              ;DVLCR OK?
015174  104003                  HLT     3               ;DVLCR INCORECT (DISREGUARD BIT5,4,1,0)
015176  005723          4$:     TST     (R3)+           ;POP POINTER TO DVSRS REG
015200  011304                  MOV     (R3),R4         ;DVSRS (EXPECT ALL 0'S)
015202  001401                  BEQ     5$              ;BR IF OK
015204  104003                  HLT     3               ;DVSRS REG NOT ALL ZEROS
015206  005723          5$:     TST     (R3)+           ;POP POINTER TO DVSRA REG
015210  011304                  MOV     (R3),R4         ;DVSRA (EXPECT ALL 0'S)
015212  001401                  BEQ     6$              ;BR IF GOOD
015214  104003                  HLT     3               ;DVSRA NOT ALL 0'S
015216  005723          6$:     TST     (R3)+           ;POP POINTER TO DVSFR
015220  011304                  MOV     (R3),R4         ;DVSFR (EXPECT ALL 1'S (THATS RIGHT))
015222  012705  177777          MOV     #177777,R5      ;SET EXPECTED
015226  020504                  CMP     R5,R4           ;EXPECETD =FOUND?
015230  001401                  BEQ     7$              ;BR IF YES
015232  104003                  HLT     3               ;DVSFR NOT ALL 1'S
015234  005723          7$:     TST     (R3)+           ;POP POINTER TO DVNSR REG
015236  005713                  TST     (R3)            ;DVNSR S/B PLUS (15=0)
015240  100001                  BPL     64$
015242  104000                  HLT     0
```

```
3366  015244  005723                  64$:    TST     (R3)+           ;POP POINTER TO RESV16 REG
3367  015246  011304                          MOV     (R3),R4         ;RESV16 (EXPECT ALL 0'S)
3368  015250  005005                          CLR     R5              ;SET EXPECTED TO 0'S
3369  015252  020504                          CMP     R5,R4           ;WELL DOES IT =1'S?
3370  015254  001401                          BEQ     8$              ;BR IF OK
3371  015256  104003                          HLT     3               ;RESV16 NOT ALL 0'S
3372  015260  104400                  8$:     SCOPE                   ;SCOPE THIS TEST;
3373
3374
3375                                          ;************************ TEST 62 *******************************
3376                                          ;*ATTACK OF THE SPECIAL FUNCTIONS REGISTER.
3377                                          ;*BEGIN CHECK OF THE DVSFR.
3378                                          ;*SUMARY OF PROC. INSTRUCTIONS
3379                                          ;*
3380                                          ;*BIT14 BIT13 BIT12  INSTRUCTION
3381                                          ;*  0     0     0    BRANCH "A"
3382                                          ;*  0     0     1    ALU OPERATION
3383                                          ;*  0     1     0    RAM OPERATION
3384                                          ;*  0     1     1    DATA TRANSFER
3385                                          ;*  1     0     0    NPR OPERATION
3386                                          ;*  1     0     1    SET/CLEAR OPERATION
3387                                          ;*  1     1     0    BCC CALCULATION
3388                                          ;*  1     1     1    BRANCH "B"
3389                                          ;*
3390                                          ;;*************************************************************************
3391
3392                                          ;************************ TEST 62 *******************************
3393                                          ;*VERIFY THAT "ROM STEP"
3394                                          ;*IS SELF-CLEARING AND THAT
3395                                          ;*THE DATA IN THE DVSFR
3396                                          ;*IS CHANGED WHEN THE ROM IS STEPPED.
3397                                          ;;*************************************************************************
3398
3399                                  ;  TEST 62
3400                                  ;--------------
3401  015262  012737  000062  001226  TST62:  MOV     #62,TSTNO
3402  015270  012737  015364  001216          MOV     #TST63,NEXT
3403  015276  104412                          MSTCLR                  ;CLEAR ALL THE DV11
3404  015300  012777  000010  164054          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
3405  015306  012705  050000                  MOV     #S.C,R5 ;PUT INSTR INTO DVSFR
3406  015312  010577  164062                  MOV     R5,@DVSFR       ;
3407  015316  027705  164056                  CMP     @DVSFR,R5       ;WAS THE DVSFR REALLY LOADED?
3408  015322  001401                          BEQ     1$              ;BR IF YES
3409  015324  104000                          HLT                     ;BAD DVSFR
3410  015326  042777  000010  164026  1$:     BIC     #BIT3,@DVSCR    ;CLEAR SOURCE SEL.
3411  015334  104415                          ROMCLK                  ;ISSUE A ROM CLOCK
3412  015336  000240                          NOP                     ;WAIST AN INTRUSTION TIME
3413  015340  032777  000002  164014          BIT     #BIT1,@DVSCR    ;DID CLK BIT CLEAR BY IT SELF?
3414  015346  001401                          BEQ     2$              ;BR IF CLK GONE
3415  015350  104000                          HLT                     ;BIT 1 OF DVSCR (ROM CLK) NOT ZERO
3416  015352  020577  164022          2$:     CMP     R5,@DVSFR       ;HAS DATA IN DVSFR CHANGED?
3417  015356  001001                          BNE     3$              ;BR IF YES
3418  015360  104000                          HLT                     ;DATA NOT CHANGED (DID CLK REALLY CLK??)
3419  015362  104400                  3$:     SCOPE                   ;SCOPE THIS TEST.
3420
3421                                          ;************************ TEST 63 *******************************
```

```
                                        ;*BASIC TEST OF DVSFR
                                        ;*TEST THAT 'BRANCH A' INSTRUCTION.
                                        ;*POINTS TESTED:
                                        ;*
                                        ;*BIT11  BIT10  BIT09  BIT08  BR "A"  BR "B"  FUNCTION
                                        ;*  0      0      0      1       L       H     PLUS 3 VOLTS
                                        ;*  0      1      0      1       L,H     H,H   DVSCR08 (=0,=1)
                                        ;*  0      1      1      1       H       H     NPR SILO NOT AVAIL.
                                        ;*  1      1      1      1       H       H     SILO FULL
                                        ;*  0      1      1      0       H       H     NXM
                                        ;
                                        ;**************************************************************

                                    ;  TEST 63
                                    ;--------------
015364  012737  000063  001226    TST63:  MOV     #63,TSTNO
015372  012737  015630  001216            MOV     #TST64,NEXT
015400  104412                            MSTCLR                  ;CLEAR DV11
015402  013700  001400                    MOV     DVSFR,R0        ;SET DVSFR POINTER TO R0
015406  013703  001370                    MOV     DVLCR,R3        ;SET DVLCR POINTER TO R3
015412  012777  000410  163742            MOV     #BIT8+BIT3,@DVSCR
015420  012710  000400              1$:   MOV     #BIT8,(R0)      ;BR-A TEST +3
015424  011002                            MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015426  012705  000002                    MOV     #BIT1,R5        ;SET EXPECTED RESULTS
015432  011304                            MOV     (R3),R4         ;READ DVLCR INTO R4
015434  042704  177774                    BIC     #^C<BIT1+BIT0>,R4
                                                                  ;CLEAR UNWANTED BITS.
015440  020504                            CMP     R5,R4           ;EXPECTED = FOUND??
015442  001401                            BEQ     2$              ;BR IF OK
015444  104006                            HLT     6               ;BR POINT WRONG
015446  012710  002400              2$:   MOV     #BIT10+BIT8,(R0)
015452  011002                            MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015454  012705  000002                    MOV     #BIT1,R5        ;SET EXPECTED RESULTS
015460  011304                            MOV     (R3),R4         ;READ DVLCR INTO R4
015462  042704  177774                    BIC     #^C<BIT1+BIT0>,R4
                                                                  ;CLEAR UNWANTED BITS.
015466  020504                            CMP     R5,R4           ;EXPECTED = FOUND??
015470  001401                            BEQ     3$              ;BR IF OK
015472  104006                            HLT     6               ;BR POINT WRONG
015474  042777  000400  163660    3$:     BIC     #BIT8,@DVSCR    ;RESET BIT 8 TO =0
015502  011002                            MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015504  012705  000003                    MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
015510  011304                            MOV     (R3),R4         ;READ DVLCR INTO R4
015512  042704  177774                    BIC     #^C<BIT1+BIT0>,R4
                                                                  ;CLEAR UNWANTED BITS.
015516  020504                            CMP     R5,R4           ;EXPECTED = FOUND??
015520  001401                            BEQ     4$              ;BR IF OK
015522  104006                            HLT     6               ;BR POINT WRONG
015524  012710  003400              4$:   MOV     #BIT10+BIT9+BIT8,(R0)
015530  011002                            MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015532  012705  000003                    MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
015536  011304                            MOV     (R3),R4         ;READ DVLCR INTO R4
015540  042704  177774                    BIC     #^C<BIT1+BIT0>,R4
                                                                  ;CLEAR UNWANTED BITS.
015544  020504                            CMP     R5,R4           ;EXPECTED = FOUND??
015546  001401                            BEQ     5$              ;BR IF OK
```

```
3478  015550  104006                        HLT     6                       ;BR POINT WRONG
3479  015552  012710  007400          5$:   MOV     #BIT11+BIT10+BIT9+BIT8,(R0)
3480  015556  011002                        MOV     (R0),R2                 ;READ DVSFR FOR PRINTOUT
3481  015560  012705  000003                MOV     #BIT1+BIT0,R5           ;SET EXPECTED RESULTS
3482  015564  011304                        MOV     (R3),R4                 ;READ DVLCR INTO R4
3483  015566  042704  177774                BIC     #^C<BIT1+BIT0>,R4
3484                                                                        ;CLEAR UNWANTED BITS.
3485  015572  020504                        CMP     R5,R4                   ;EXPECTED = FOUND??
3486  015574  001401                        BEQ     6$                      ;BR IF OK
3487  015576  104006                        HLT     6                       ;BR POINT WRONG
3488  015600  012710  003000          6$:   MOV     #BIT10+BIT9,(R0)                ;NXM
3489  015604  011002                        MOV     (R0),R2                 ;READ DVSFR FOR PRINTOUT
3490  015606  012705  000003                MOV     #BIT1+BIT0,R5           ;SET EXPECTED RESULTS
3491  015612  011304                        MOV     (R3),R4                 ;READ DVLCR INTO R4
3492  015614  042704  177774                BIC     #^C<BIT1+BIT0>,R4
3493                                                                        ;CLEAR UNWANTED BITS.
3494  015620  020504                        CMP     R5,R4                   ;EXPECTED = FOUND??
3495  015622  001401                        BEQ     7$                      ;BR IF OK
3496  015624  104006                        HLT     6                       ;BR POINT WRONG
3497  015626  104400                  7$:   SCOPE                           ;SCOPE TEST
3498
3499                                        ;********************** TEST 64 ****************************
3500                                        ;*TEST OF BRANCH B''
3501                                        ;*TEST THAT POINT 16 (GROUND)
3502                                        ;*MAKES LCR BIT1=1 AND BIT0=1.
3503                                        ;;**************************************************************
3504
3505                                  ;  TEST 64
3506                                  ;--------------
3507  015630  012737  000064  001226  TST64:  MOV     #64,TSTNO
3508  015636  012737  015706  001216          MOV     #TST65,NEXT
3509  015644  104412                        MSTCLR                          ;CLEAR DV11
3510  015646  012777  000010  163506        MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
3511  015654  012777  077400  163516        MOV     #BRB+BIT11+BIT10+BIT9+BIT8,@DVSFR
3512                                                                        ;BR-B ''GROUND''?
3513  015662  017702  163512                MOV     @DVSFR,R2               ;READ DVSFR INTO R2
3514  015666  012705  000003                MOV     #BIT1+BIT0,R5           ;SET EXPECTED RESULTS
3515  015672  017704  163472                MOV     @DVLCR,R4               ;READ REAL RESULTS
3516  015676  020504                        CMP     R5,R4                   ;SAME??
3517  015700  001401                        BEQ     1$                      ;
3518  015702  104006                        HLT     6                       ;BR TEST POINT WRONG
3519  015704  104400                  1$:   SCOPE                           ;SCOPE THIS TEST
3520
3521                                        ;********************** TEST 65 ****************************
3522                                        ;*TEST OF BRANCH B
3523                                        ;*CHECKING ''DEFAULT'' STATES OF THE DV11 SIGNALS.
3524                                        ;*BIT11 BIT10   BIT09   BIT08   FUNCTION
3525                                        ;*  1     0       0       0     DATA NOT AVAIL.
3526                                        ;*  1     0       0       1     REQUEST BUS
3527                                        ;*  1     0       1       0     MEMORY PARITY ERROR
3528                                        ;*  1     1       1       1     WRITE INHIBIT
3529                                        ;;**************************************************************
3530
3531                                  ;  TEST 65
3532                                  ;--------------
3533  015706  012737  000065  001226  TST65:  MOV     #65,TSTNO
```

```
3534  015714  012737  016074  001216          MOV     #TST66,NEXT
3535  015722  104412                          MSTCLR
3536  015724  013700  001400                  MOV     DVSFR,R0
3537  015730  013703  001370                  MOV     DVLCR,R3
3538  015734  012777  000010  163420          MOV     #BIT3,@DVSCR
3539  015742  012710  074000          1$:     MOV     #BRB+BIT11,(R0)
3540  015746  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
3541  015750  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
3542  015754  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
3543  015756  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
3544                                                                  ;CLEAR UNWANTED BITS.
3545  015762  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
3546  015764  001401                          BEQ     2$              ;BR IF OK
3547  015766  104006                          HLT     6               ;BR POINT WRONG
3548  015770  012710  074400          2$:     MOV     #BRB+BIT11+BIT8,(R0)
3549  015774  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
3550  015776  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
3551  016002  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
3552  016004  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
3553                                                                  ;CLEAR UNWANTED BITS.
3554  016010  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
3555  016012  001401                          BEQ     3$              ;BR IF OK
3556  016014  104006                          HLT     6               ;BR POINT WRONG
3557  016016  012710  075000          3$:     MOV     #BRB+BIT11+BIT9,(R0)
3558  016022  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
3559  016024  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
3560  016030  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
3561  016032  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
3562                                                                  ;CLEAR UNWANTED BITS.
3563  016036  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
3564  016040  001401                          BEQ     4$              ;BR IF OK
3565  016042  104006                          HLT     6               ;BR POINT WRONG
3566  016044  012710  077000          4$:     MOV     #BRB+BIT11+BIT10+BIT9,(R0)
3567  016050  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
3568  016052  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
3569  016056  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
3570  016060  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
3571                                                                  ;CLEAR UNWANTED BITS.
3572  016064  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
3573  016066  001401                          BEQ     5$              ;BR IF OK
3574  016070  104006                          HLT     6               ;BR POINT WRONG
3575  016072  104400                  5$:     SCOPE
3576
3577
3578
3579                                          ;************************* TEST 66 *****************************
3580                                          ;*BASIC TEST OF THE
3581                                          ;*''SET/CLEAR INSTRUCTION.
3582                                          ;*TEST THAT THE SET/CLEAR CAN DO:
3583                                          ;*CLEAR DVSCR 08
3584                                          ;*SET DVSCR10
3585                                          ;*SET RECEIVER INTERUPT (DVSCR07)
3586                                          ;;***************************************************************
3587
3588                                  ;  TEST 66
3589                                  ;--------------
```

```
3590  016074  012737  000066  001226  TST66:  MOV     #66,TSTNO
3591  016102  012737  016266  001216          MOV     #TST67,NEXT
3592  016110  012737  016140  001220          MOV     #1$,LOCK
3593  016116  104412                          MSTCLR                  ;CLEAR DV11
3594  016120  052777  000400  163234          BIS     #BIT8,@DVSCR    ;SET BIT8.
3595  016126  052777  000010  163226          BIS     #BIT3,@DVSCR    ;SET SOURCE SEL.
3596  016134  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER ADDRESS IN R0
3597  016140  012710  050016          1$:     MOV     #S.C+BIT3+BIT2+BIT1,(R0)
3598  016144  012705  000010                  MOV     #BIT3,R5        ;DO SET/CLEAR -CLEAR BIT 8 OF DVSCR
3599  016150  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
3600  016152  104415                          ROMCLK                  ;CYCLE THE ROM
3601  016154  017704  163202                  MOV     @DVSCR,R4       ;READ DVSCR INTO 'FOUND LOC.
3602  016160  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
3603  016162  001401                          BEQ     64$             ;BR IF DVSCR OK
3604  016164  104006                          HLT     6               ;ROM FAILED TO EXECUTE
3605  016166  104401          64$:    SCOP1                           ;LOCK ON THIS SUB-TEST? SW09=1?
3606  016170  012737  016176  001220          MOV     #3$,LOCK        ;SET FOR RETURN IF SW09=1
3607  016176  052705  000200          3$:     BIS     #BIT7,R5        ;SET EXPECTED (SCR BIT 7=1)
3608  016202  012710  050013                  MOV     #S.C+BIT3+BIT1+BIT0,(R0)
3609  016206  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
3610  016210  104415                          ROMCLK                  ;CYCLE THE ROM
3611  016212  017704  163144                  MOV     @DVSCR,R4       ;READ DVSCR INTO 'FOUND LOC.
3612  016216  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
3613  016220  001401                          BEQ     65$             ;BR IF DVSCR OK
3614  016222  104006                          HLT     6               ;ROM FAILED TO EXECUTE
3615  016224  104401          65$:    SCOP1                           ;LOCK ON THIS SUB-TEST? SW09=1?
3616  016226  012737  016234  001220          MOV     #4$,LOCK
3617  016234  052705  002000          4$:     BIS     #BIT10,R5       ;ALTER EXPECTED ADDRESS
3618  016240  012710  050012                  MOV     #S.C+BIT3+BIT1,(R0)
3619                                                                  ;DO A SET/CLEAR SET DVSCR BIT 10
3620  016244  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
3621  016246  104415                          ROMCLK                  ;CYCLE THE ROM
3622  016250  017704  163106                  MOV     @DVSCR,R4       ;READ DVSCR INTO 'FOUND LOC.
3623  016254  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
3624  016256  001401                          BEQ     66$             ;BR IF DVSCR OK
3625  016260  104006                          HLT     6               ;ROM FAILED TO EXECUTE
3626  016262  104401          66$:    SCOP1                           ;LOCK ON THIS SUB-TEST? SW09=1?
3627  016264  104400                          SCOPE                   ;SCOPE THIS TEST
3628
3629
3630                                          ;************************* TEST 67 *******************************
3631                                          ;*BASIC TEST OF THE
3632                                          ;*'SET/CLEAR INSTRUCTION.
3633                                          ;*TEST THAT THE SET/CLEAR CAN:
3634                                          ;*
3635                                          ;*BIT 14  BIT12  BIT03  BIT02  BIT01  BIT00  FUNCTION
3636                                          ;*  1       1       1      0      0      0     SET RICR 15
3637                                          ;*  1       1       1      0      0      1     SET RICR 14
3638                                          ;*  1       1       1      1      0      0     SET RICR 13
3639                                          ;*  1       1       1      1      0      1     SET RICR 12
3640                                          ;*
3641                                          ;;*****************************************************************
3642
3643                                  ;  TEST 67
3644                                  ;--------------
3645  016266  012737  000067  001226  TST67:  MOV     #67,TSTNO
```

```
3646  016274  012737  016510  001216          MOV     #TST70,NEXT
3647  016302  104412                          MSTCLR                  ;CLEAR DV11
3648  016304  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER TO R0
3649  016310  012777  000010  163044          MOV     #BIT3,@DVSCR    ;SET SORCE SEL
3650  016316  012705  100000                  MOV     #BIT15,R5       ;SET EXPECTED RESULTS
3651  016322  012710  050010                  MOV     #S.C+BIT3,(R0)
3652                                                                  ;SET/CLEAR DVRICR 15
3653  016326  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
3654  016330  104415                          ROMCLK                  ;CYCLE THE ROM
3655  016332  017704  163030                  MOV     @DVRIC,R4       ;READ DVRIC INTO 'FOUND' LOC.
3656  016336  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
3657  016340  001401                          BEQ     64$             ;BR IF DVSCR OK
3658  016342  104006                          HLT     6               ;ROM FAILED TO EXECUTE
3659  016344  104401              64$:        SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
3660  016346  104412                          MSTCLR                  ;CLEAR DV11
3661  016350  012777  000010  163004          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
3662  016356  012705  040000                  MOV     #BIT14,R5       ;SET EXPECTED RESULTS
3663  016362  012710  050011                  MOV     #S.C+BIT3+BIT0,(R0)
3664                                                                  ;SET/CLEAR DVRICR 14
3665  016366  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
3666  016370  104415                          ROMCLK                  ;CYCLE THE ROM
3667  016372  017704  162770                  MOV     @DVRIC,R4       ;READ DVRIC INTO 'FOUND' LOC.
3668  016376  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
3669  016400  001401                          BEQ     65$             ;BR IF DVSCR OK
3670  016402  104006                          HLT     6               ;ROM FAILED TO EXECUTE
3671  016404  104401              65$:        SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
3672  016406  104412                          MSTCLR                  ;CLEAR DV11
3673  016410  012777  000010  162744          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
3674  016416  012705  020000                  MOV     #BIT13,R5       ;SET EXPECTED RESULTS
3675  016422  012710  050014                  MOV     #S.C+BIT3+BIT2,(R0)
3676                                                                  ;SET/CLEAR DVRICR 13
3677  016426  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
3678  016430  104415                          ROMCLK                  ;CYCLE THE ROM
3679  016432  017704  162730                  MOV     @DVRIC,R4       ;READ DVRIC INTO 'FOUND' LOC.
3680  016436  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
3681  016440  001401                          BEQ     66$             ;BR IF DVSCR OK
3682  016442  104006                          HLT     6               ;ROM FAILED TO EXECUTE
3683  016444  104401              66$:        SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
3684  016446  104412                          MSTCLR                  ;CLEAR DV11
3685  016450  012777  000C10  162704          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
3686  016456  012705  010000                  MOV     #BIT12,R5       ;SET EXPECTED RESULTS
3687  016462  012710  050015                  MOV     #S.C+BIT3+BIT2+BIT0,(R0)
3688                                                                  ;SET/CLEAR DVRICR 12
3689  016466  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
3690  016470  104415                          ROMCLK                  ;CYCLE THE ROM
3691  016472  017704  162670                  MOV     @DVRIC,R4       ;READ DVRIC INTO 'FOUND' LOC.
3692  016476  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
3693  016500  001401                          BEQ     67$             ;BR IF DVSCR OK
3694  016502  104006                          HLT     6               ;ROM FAILED TO EXECUTE
3695  016504  104401              67$:        SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
3696  016506  104400                          SCOPE                   ;SCOPE THIS TEST
3697
3698
3699                                          ;************************* TEST 70 *******************************
3700                                          ;*BASIC TEST OF DVSFR.
3701                                          ;*TEST OF ''SET/CLEAR'' AND
```

```
                                        ;*'BRANCH A'' AND 'BRANCH B'' FUNCTIONS.
                                        ;:*****************************************************************
                                ;  TEST 70
                                ;--------------
016510  012737  000070  001226 TST70:  MOV     #70,TSTNO
016516  012737  017172  001216         MOV     #TST71,NEXT
016524  012737  016546  001220         MOV     #1$,LOCK
016532  104412                        MSTCLR                  ;CLEAR DV11
016534  012777  000010  162620         MOV     #BIT3,@DVSCR    ;SET SOURCE SELECT
016542  013700  001400                 MOV     DVSFR,R0        ;SET DVSFR POINTER INTO R0

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR ALU BIT02
016546  004237  017136          1$:    JSR     R2,10$          ;GOTO SUBROUTINE.
016552  000024                         BIT4+BIT2               ;POINT SET/CLEARED
016554  006000                         BIT11+BIT10             ;BR TEST POINT
016556  000002                         BIT1                    ;EXPECTED RESULTS IN DVLCR
016560  004237  017136                 JSR     R2,10$          ;GOSUB
016564  000025                         BIT4+BIT2+BIT0          ;POINT SET/CLEARED
016566  006000                         BIT11+BIT10             ;BR TEST POINT
016570  000003                         BIT1+BIT0               ;EXPECTED RESUTS IN DVLCR
016572  104401                         SCOP1                   ;SW09=1??

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT00
016574  012737  016602  001220         MOV     #2$,LOCK        ;SET RETURN IF SW09=1
016602  004237  017136          2$:    JSR     R2,10$  ;GOTO THE SUBROUTINE
016606  000047                         BIT5+BIT2+BIT1+BIT0                     ;POINT SET/CLEARED [SET]
016610  070000                         BRB                     ;BR TEST POINT
016612  000001                         BIT0                    ;DVLCR EXPECTED
016614  004237  017136                 JSR     R2,10$          ;GOTO SUB ROUTINE
016620  000043                         BIT5+BIT1+BIT0                  ;POINT SET/CLEARED [CLEARED]
016622  070000                         BRB                     ;BR TEST POINT
016624  000003                         BIT1+BIT0               ;EXPECTED RESULTS
016626  104401                 65$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT01
016630  012737  016636  001220         MOV     #3$,LOCK        ;SET RETURN IF SW09=1
016636  004237  017136          3$:    JSR     R2,10$  ;GOTO THE SUBROUTINE
016642  000045                         BIT5+BIT2+BIT0                  ;POINT SET/CLEARED [SET]
016644  070400                         BRB+BIT8                ;BR TEST POINT
016646  000001                         BIT0                    ;DVLCR EXPECTED
016650  004237  017136                 JSR     R2,10$          ;GOTO SUB ROUTINE
016654  000041                         BIT5+BIT0                       ;POINT SET/CLEARED [CLEARED]
016656  070400                         BRB+BIT8                ;BR TEST POINT
016660  000003                         BIT1+BIT0               ;EXPECTED RESULTS
016662  104401                 67$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT02
016664  012737  016672  001220         MOV     #4$,LOCK        ;SET RETURN IF SW09=1
016672  004237  017136          4$:    JSR     R2,10$  ;GOTO THE SUBROUTINE
016676  000046                         BIT5+BIT2+BIT1                  ;POINT SET/CLEARED [SET]
016700  071000                         BRB+BIT9                ;BR TEST POINT
```

```
3758  016702  000001                        BIT0                    ;DVLCR EXPECTED
3759  016704  004237  017136                JSR     R2,10$          ;GOTO SUB ROUTINE
3760  016710  000042                        BIT5+BIT1                       ;POINT SET/CLEARED [CLEARED]
3761  016712  071000                        BRB+BIT9                        ;BR TEST POINT
3762  016714  000003                        BIT1+BIT0               ;EXPECTED RESULTS
3763  016716  104401              69$:      SCOP1                   ;SWR 09=1?
3764
3765                                        ;*TEST SET/CLEAR FUNCTION
3766                                        ;*FOR RAM OUTPUT BIT03
3767  016720  012737  016726  001220        MOV     #6$,LOCK        ;SET RETURN IF SW09=1
3768  016726  004237  017136        6$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
3769  016732  000044                        BIT5+BIT2                       ;POINT SET/CLEARED [SET]
3770  016734  071400                        BRB+BIT9+BIT8                   ;BR TEST POINT
3771  016736  000001                        BIT0                    ;DVLCR EXPECTED
3772  016740  004237  017136                JSR     R2,10$          ;GOTO SUB ROUTINE
3773  016744  000040                        BIT5                    ;POINT SET/CLEARED [CLEARED]
3774  016746  071400                        BRB+BIT9+BIT8                   ;BR TEST POINT
3775  016750  000003                        BIT1+BIT0               ;EXPECTED RESULTS
3776  016752  104401              71$:      SCOP1                   ;SWR 09=1?
3777
3778                                        ;*TEST SET/CLEAR FUNCTION
3779                                        ;*FOR RAM OUTPUT BIT04
3780  016754  012737  016762  001220        MOV     #7$,LOCK        ;SET RETURN IF SW09=1
3781  016762  004237  017136        7$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
3782  016766  000207                        BIT7+BIT2+BIT1+BIT0                     ;POINT SET/CLEARED [SET]
3783  016770  072000                        BRB+BIT10                       ;BR TEST POINT
3784  016772  000001                        BIT0                    ;DVLCR EXPECTED
3785  016774  004237  017136                JSR     R2,10$          ;GOTO SUB ROUTINE
3786  017000  000203                        BIT7+BIT1+BIT0                  ;POINT SET/CLEARED [CLEARED]
3787  017002  072000                        BRB+BIT10                       ;BR TEST POINT
3788  017004  000003                        BIT1+BIT0               ;EXPECTED RESULTS
3789  017006  104401              73$:      SCOP1                   ;SWR 09=1?
3790
3791                                        ;*TEST SET/CLEAR FUNCTION
3792                                        ;*FOR RAM OUTPUT BIT05
3793  017010  012737  017016  001220        MOV     #8$,LOCK        ;SET RETURN IF SW09=1
3794  017016  004237  017136        8$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
3795  017022  000205                        BIT7+BIT2+BIT0                  ;POINT SET/CLEARED [SET]
3796  017024  072400                        BRB+BIT10+BIT8                  ;BR TEST POINT
3797  017026  000001                        BIT0                    ;DVLCR EXPECTED
3798  017030  004237  017136                JSR     R2,10$          ;GOTO SUB ROUTINE
3799  017034  000201                        BIT7+BIT0                       ;POINT SET/CLEARED [CLEARED]
3800  017036  072400                        BRB+BIT10+BIT8                  ;BR TEST POINT
3801  017040  000003                        BIT1+BIT0               ;EXPECTED RESULTS
3802  017042  104401              75$:      SCOP1                   ;SWR 09=1?
3803
3804                                        ;*TEST SET/CLEAR FUNCTION
3805                                        ;*FOR RAM OUTPUT BIT06
3806  017044  012737  017052  001220        MOV     #9$,LOCK        ;SET RETURN IF SW09=1
3807  017052  004237  017136        9$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
3808  017056  000206                        BIT7+BIT2+BIT1                  ;POINT SET/CLEARED [SET]
3809  017060  073000                        BRB+BIT10+BIT9                  ;BR TEST POINT
3810  017062  000001                        BIT0                    ;DVLCR EXPECTED
3811  017064  004237  017136                JSR     R2,10$          ;GOTO SUB ROUTINE
3812  017070  000202                        BIT7+BIT1                       ;POINT SET/CLEARED [CLEARED]
3813  017072  073000                        BRB+BIT10+BIT9                  ;BR TEST POINT
```

```
3814  017074  000003                    BIT1+BIT0                ;EXPECTED RESULTS
3815  017076  104401          77$:      SCOP1                    ;SWR 09=1?
3816
3817                                    ;*TEST SET/CLEAR FUNCTION
3818                                    ;*FOR RAM OUTPUT BIT07
3819  017100  012737  017106  001220    MOV     #101$,LOCK       ;SET RETURN IF SW09=1
3820  017106  004237  017136  101$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
3821  017112  000204                    BIT7+BIT2                        ;POINT SET/CLEARED [SET]
3822  017114  073400                    BRB+BIT10+BIT9+BIT8                      ;BR TEST POINT
3823  017116  000001                    BIT0                     ;DVLCR EXPECTED
3824  017120  004237  017136            JSR     R2,10$           ;GOTO SUB ROUTINE
3825  017124  000200                    BIT7                     ;POINT SET/CLEARED [CLEARED]
3826  017126  073400                    BRB+BIT10+BIT9+BIT8                      ;BR TEST POINT
3827  017130  000003                    BIT1+BIT0                ;EXPECTED RESULTS
3828  017132  104401          79$:      SCOP1                    ;SWR 09=1?
3829  017134  104400                    SCOPE                    ;SCOPE THE TEST
3830  017136  012710  050000    10$:    MOV     #S.C,(R0)        ;SET/CLEAR INSTR
3831  017142  010201                    MOV     R2,R1            ;SAVE JSR PC ADDRESS
3832  017144  052210                    BIS     (R2)+,(R0)       ;LOAD POINT SET/CLEARED
3833  017146  104415                    ROMCLK                   ;CYCLE THE ROM
3834  017150  012210                    MOV     (R2)+,(R0)       ;LOAD BR TEST POINT
3835  017152  011003                    MOV     (R0),R3          ;READ DVSFR INTO R3
3836  017154  012205                    MOV     (R2)+,R5         ;LOAD EXPECTED INTO R5
3837  017156  017704  162206            MOV     @DVLCR,R4        ;READ DVLCR INTO FOUND LOC.
3838  017162  020504                    CMP     R5,R4            ;EXPECTED=FOUND?
3839  017164  001401                    BEQ     11$              ;BR IF YES
3840  017166  104005                    HLT     5                ;DVLCR WRONG BR RESULTS.
3841  017170  000202          11$:      RTS     R2               ;RETURN
3842
3843                                    ;************************ TEST 71 *******************************
3844                                    ;*TEST OF 'RECEIVER CHARACTER SILO''
3845                                    ;*THRU THE USE OF THE DVSFR REG.
3846                                    ;*TEST THE FILLING THE SILO PRODUCES ''SILO FULL''
3847                                    ;*ON EXACTLY THE 128 LOAD.
3848                                    ;*SET/CLEAR IS USED TO STUFF SILO AND BRANCH A IS USED TO TEST SILO.
3849                                    ;*SET/CLEAR ''SILO IN'' AND SET/CLEAR ''SILO OUT'' ARE EXERCISED TOO.
3850                                    ;:***************************************************************************
3851
3852                                  ;  TEST 71
3853                                  ;--------------
3854  017172  012737  000071  001226  TST71:  MOV     #71,TSTNO
3855  017200  012737  017414  001216          MOV     #TST72,NEXT
3856  017206  104412                         MSTCLR                   ;CLEAR DV11
3857  017210  012700  000177                 MOV     #127.,R0         ;SET R0 TO 1 LESS THAN FULL SILO
3858  017214  012777  000010  162140         MOV     #BIT3,@DVSCR     ;SET SOURCE SEL
3859  017222  012705  000003                 MOV     #BIT1+BIT0,R5    ;SET EXPECTED RESULTS INTO R5
3860  017226  012777  050021  162144  1$:    MOV     #S.C+BIT4+BIT0,@DVSFR
3861  017234  104415                         ROMCLK                   ;S/C ''SILO IN''
3862  017236  012777  007400  162134         MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
3863                                                                  ;BR-A ''SILO FULL''?
3864  017244  017702  162130                 MOV     @DVSFR,R2        ;SAVE CONTENTS OF DVSFR FOR ERROR PRINTOUT
3865  017250  017704  162114                 MOV     @DVLCR,R4        ;READ DVLCR FOR RESULTS
3866  017254  020504                         CMP     R5,R4            ;ARE BR TEST POINTS CORRECT?
3867  017256  001401                         BEQ     64$              ;BR IF YES
3868  017260  104006                         HLT     6                ;BR TEST POINTS WRONG (BIT1 OR 0)
3869  017262  005300          64$:           DEC     R0               ;IS SILO FULL-1 YET?
```

```
3870  017264  001360                          BNE     1$                  ;BR IF NOT 127 TIMES YET
3871  017266  012777  050021  162104  2$:     MOV     #S.C+BIT4+BIT0,@DVSFR
3872                                                                      ;S/C ''SILO IN''
3873  017274  104415                          ROMCLK                      ;
3874  017276  000240                          NOP                         ;WAIST INSTRUCTION TIME
3875  017300  012777  007400  162072          MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
3876                                                                      ;BR-A ''SILO FULL''?
3877  017306  017702  162066                  MOV     @DVSFR,R2           ;SAVE DVSFR
3878  017312  017704  162052                  MOV     @DVLCR,R4           ;READ BR TEST POINTS
3879  017316  042705  000001                  BIC     #BIT0,R5            ;ALTER EXPECTED RESULTS
3880  017322  020504                          CMP     R5,R4               ;BR TEST POINTS OK??
3881  017324  001401                          BEQ     3$                  ;BR IF YES
3882  017326  104006                          HLT     6                   ;BR TEST POINTS WRONG
3883  017330  012777  050020  162042  3$:     MOV     #S.C+BIT4,@DVSFR
3884  017336  104415                          ROMCLK                      ;S/C ''SILO OUT''
3885  017340  000240                          NOP                         ;WAIST INSTR TIME
3886  017342  012777  007400  162030          MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
3887                                                                      ;BR-A ''SILO FULL''?
3888  017350  005002                          CLR     R2                  ;DEALAY AT LEAST 32US
3889  017352  032777  000001  162010  4$:     BIT     #BIT0,@DVLCR        ;IS SILO *NOT FULL*??
3890  017360  001003                          BNE     5$                  ;BR IF OK.
3891  017362  062702  000001                  ADD     #1,R2               ;DELAY.....
3892  017366  001371                          BNE     4$                  ;GOTO 4$
3893  017370  017702  162004          5$:     MOV     @DVSFR,R2           ;SAVE DVSFR
3894  017374  017704  161770                  MOV     @DVLCR,R4           ;READ BR TEST POINTS
3895  017400  052705  000001                  BIS     #BIT0,R5            ;SET EXPECTED RESULTS
3896  017404  020504                          CMP     R5,R4               ;OK??
3897  017406  001401                          BEQ     6$                  ;YES
3898  017410  104006                          HLT     6                   ;SILO  STILL FULL.
3899  017412  104400                  6$:     SCOPE                       ;SCOPE TEST
3900
3901                                          ;************************ TEST 72 *******************************
3902                                          ;*TEST THAT AFTER AN INIT
3903                                          ;*THAT ''RECV CHARACTER WAITING''
3904                                          ;*IS FALSE (HIGH) AND THEN VERIFY
3905                                          ;*THAT WHEN ''SILO IN'' IS ASSERTED THAT
3906                                          ;*THAT ''RCVED CHARACTER WAITING'' IS TRUE (LOW)
3907                                          ;*AND MAKES ''BRANCH A'' TRUE,
3908                                          ;:*************************************************************************
3909
3910                                  ;  TEST 72
3911                                  ;--------------
3912  017414  012737  000072  001226  TST72:  MOV     #72,TSTNO
3913  017422  012737  017620  001216          MOV     #TST73,NEXT
3914  017430  104412                          MSTCLR                      ;CLEAR DV11
3915  017432  012777  000010  161722          MOV     #BIT3,@DVSCR        ;SET SOURCE SEL
3916  017440  012705  000003                  MOV     #BIT1+BIT0,R5       ;SET EXPECTED RESULTS
3917  017444  012702  001400                  MOV     #BIT9+BIT8,R2       ;BR-A ''RCVD CHAR WAITING''?
3918  017450  010277  161724                  MOV     R2,@DVSFR           ;LOAD DV INSTR
3919  017454  017704  161710                  MOV     @DVLCR,R4           ;READ TEST POINTS
3920  017460  020504                          CMP     R5,R4               ;OK??
3921  017462  001401                          BEQ     64$                 ;YES
3922  017464  104006                          HLT     6                   ;TEST POINT RECV CHAR WAITING WRONG
3923  017466  012702  050021          64$:    MOV     #S.C+BIT4+BIT0,R2
3924                                                                      ;S/C ''SILO IN''
3925  017472  010277  161702                  MOV     R2,@DVSFR           ;LOAD INSTR
```

```
3926  017476  104415                          ROMCLK                  ;CLOCK
3927  017500  005004                          CLR     R4              ;PREPARE COUNTER
3928  017502  012702  001400                  MOV     #BIT9+BIT8,R2   ;BR-A RCV CHAR WAITING
3929                                                                  ;BR-A 'RCVD CHAR WAITING'?
3930  017506  010277  161666                  MOV     R2,@DVSFR       ;LOAD INSTR
3931  017512  012705  000002                  MOV     #BIT1,R5        ;SET GOOD RESULTS
3932  017516  032777  000001  161644  1$:     BIT     #BIT0,@DVLCR    ;TEST DV11 BR POINT
3933  017524  001403                          BEQ     2$              ;BR IF OK
3934  017526  062704  000001                  ADD     #1,R4           ;DELAY
3935  017532  001371                          BNE     1$              ;GOTO 1$
3936  017534  017704  161630          2$:     MOV     @DVLCR,R4       ;READ DV11 BR POINT
3937  017540  020504                          CMP     R5,R4           ;
3938  017542  001401                          BEQ     3$
3939  017544  104006                          HLT     6               ;BR POINT RCV CHAR WAITING WRONG
3940                                          ;*TEST THAT SETTING DVSCR07
3941                                          ;*INHIBITS RCV CHAR WAITING FROM APPEARING
3942                                          ;*TRUE; AND THAT CLEARING
3943                                          ;*DVSCR07 MAKES IT APPEAR TRUE AGAIN.
3944
3945  017546  012705  000003          3$:     MOV     #BIT1+BIT0,R5   ;LOAD EXPECTED
3946  017552  052777  001200  161602          BIS     #BIT9+BIT7,@DVSCR       ;SET RECV INTER
3947  017560  017704  161604                  MOV     @DVLCR,R4       ;READ DV BR POINTS
3948  017564  020504                          CMP     R5,R4           ;
3949  017566  001401                          BEQ     4$              ;
3950  017570  104006                          HLT     6               ;BR TEST POINTS WRONG
3951  017572  042705  000001          4$:     BIC     #BIT0,R5        ;RESET EXPECTED RESULTS
3952  017576  042777  000200  161556          BIC     #BIT7,@DVSCR    ;CLEAR RECV INT
3953  017604  017704  161560                  MOV     @DVLCR,R4       ;READ BR POINTS
3954  017610  020504                          CMP     R5,R4           ;
3955  017612  001401                          BEQ     5$              ;
3956  017614  104006                          HLT     6               ;BR TEST POINTS WRONG
3957  017616  104400                  5$:     SCOPE                   ;SCOPE THIS TEST
3958
3959
3960                                          ;************************ TEST 73 *******************************
3961                                          ;*BASIC TEST OF THE 'DATA TRANSFER INSTRUCTION'
3962                                          ;*BITS 07,06,05,04 OF DVSFR INDICATE THE SOURCE
3963                                          ;*BITS 03,02,01,00 OF DVSFR INDICATE THE DESTINATION.
3964                                          ;;*************************************************************************
3965
3966                                  ;  TEST 73
3967                                  ;--------------
3968  017620  012737  000073  001226  TST73:  MOV     #73,TSTNO
3969  017626  012737  020160  001216          MOV     #TST74,NEXT
3970  017634  012737  017734  001220          MOV     #1$,LOCK
3971  017642  104412                          MST                     ;CLEAR DV11
3972  017644  012777  000010  161510          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
3973  017652  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER INTO R0
3974
3975                                          ;*TEST TO XFR SOURCE REGISTERS TO THE DVRIC
3976                                          ;*REGISTER VERIFYING THAT THE FOLLOWING REGISTERS
3977                                          ;*ARE CLEARED AND THAT THE XFR BUS IS CLEAR AFTER
3978                                          ;*A MSTCLR.
3979                                          ;*REGISTER      FUNCTION
3980                                          ;* 0000          GROUND
3981                                          ;* 0001          GROUND
```

```
                                        ;* 0010          GROUND
                                        ;* 0011          GROUND
                                        ;* 0100          GROUND
                                        ;* 0101          MASTER SCAN 0-3/0-3
                                        ;* 0110          ALU RESULT 8-11/0-3
                                        ;* 0111          ALU RESULT 5-7/0-2
                                        ;* 1000          LOW BYTE=B REG 8-15 ; HIGH BYTE=GRND
                                        ;* 1001          LO BYTE=NPR OUT ; HI BYTE=CDC REG
                                        ;* 1010          RAM OUTPUT 0-2/8-10
                                        ;* 1011          RAM OUTPUT
                                        ;* 1101          NPR INPUT REGISTER
                                        ;* 1110          BCC REGISTER
                                        ;* 1111          ALU RESULT REGISTER

017656  005005                          CLR     R5              ;SET EXPECTED TO 0
017660  012702  030000                  MOV     #XFR,R2         ;SET DATA XFR INSTR.
017664  052702  000006                  BIS     #BIT2+BIT1,R2   ;SET DESTINATION TO DVRIC REG.
017670  005003                          CLR     R3              ;ZERO SOURCE REG POINTER
017672  042702  000360          65$:    BIC     #BIT7+BIT6+BIT5+BIT4,R2
017676  050302                          BIS     R3,R2           ;SET SOURCE REGISTER
017700  010210                          MOV     R2,(R0) ;LOAD SFR WITH XFR INSTR
017702  104415                          ROMCLK                  ;EXECUTE INSTR
017704  017704  161456                  MOV     @DVRIC,R4       ;READ SOURCE REGISTER
017710  001401                          BEQ     66$             ;BR IF IT WAS ZERO
017712  104006                          HLT     6               ;SOURCE REGISTER IN SFR NOT ZERO
017714  062703  000020          66$:    ADD     #BIT4,R3        ;UPDATE SOURCE REGISTER
017720  022703  000300                  CMP     #300,R3         ;DON'T DO SILO REGISTER!!
017724  001773                          BEQ     66$             ;GET NEXT REG IF THIS IS SILO.
017726  032703  000360                  BIT     #BIT7+BIT6+BIT5+BIT4,R3
017732  001357                          BNE     65$             ;BR IF MORE TO DO.

                                        ;*TEST OF SET RAM OUTPUT BIT0
                                        ;*AND THE USE OF THE DATA XFER INSTR.
                                        ;*PLACE RAM BIT0 INTO THE DVRIC REG
017734  012705  000400          1$:     MOV     #BIT8,R5
017740  012703  030000                  MOV     #XFR,R3 ;-DATA XFER-
017744  052703  000246                  BIS     #BIT7+BIT5+BIT2+BIT1,R3
017750  004237  020100                  JSR     R2,10$          ;S= RAM OUTPUT 0-2. D= DVRIC
017754  000047                          BIT5+BIT2+BIT1+BIT0
017756  000043                          BIT5+BIT1+BIT0

                                        ;*TEST TO SET RAM OUTPUT DATA BIT3
                                        ;*AND THE USE OF THE 'DATA TRANSFER'' INSTRUCTION
                                        ;*TO PLACE BIT3 INTO THE DVRIC REGISTER.
017760  012737  017766  001220          MOV     #3$,LOCK        ;SET RETURN IF SW09=1
017766  012705  000010          3$:     MOV     #BIT3,R5        ;SET EXPECTED DATA
017772  012703  030000                  MOV     #XFR,R3 ;-DATA XFER-
017776  052703  000266                  BIS     #BIT7+BIT5+BIT4+BIT2+BIT1,R3
020002  004237  020100                  JSR     R2,10$          ;S= RAM OUTPUT. D=DVRIC
020006  000044                          BIT5+BIT2
020010  000040                          BIT5

                                        ;*TEST TO SET RAM OUTPUT DATA BIT4
                                        ;*AND THE USE OF THE 'DATA TRANSFER'' INSTRUCTION
                                        ;*TO PLACE BIT4 INTO THE DVRIC REGISTER.
020012  012737  020020  001220          MOV     #4$,LOCK        ;SET RETURN IF SW09=1
```

```
4038  020020  012705  000020          4$:     MOV     #BIT4,R5        ;SET EXPECTED DATA
4039  020024  012703  030000                  MOV     #XFR,R3 ;-DATA XFER-
4040  020030  052703  000266                  BIS     #BIT7+BIT5+BIT4+BIT2+BIT1,R3
4041  020034  004237  020100                  JSR     R2,10$          ;S= RAM OUTPUT. D=DVRIC
4042  020040  000207                          BIT7+BIT2+BIT1+BIT0
4043  020042  000203                          BIT7+BIT1+BIT0
4044
4045                                          ;*TEST TO SET RAM OUTPUT DATA BIT7
4046                                          ;*AND THE USE OF THE 'DATA TRANSFER' INSTRUCTION
4047                                          ;*TO PLACE BIT7 INTO THE DVRIC REGISTER.
4048  020044  012737  020052  001220          MOV     #6$,LOCK        ;SET RETURN IF SW09=1
4049  020052  012705  000200          6$:     MOV     #BIT7,R5        ;SET EXPECTED DATA
4050  020056  012703  030000                  MOV     #XFR,R3 ;-DATA XFER-
4051  020062  052703  000266                  BIS     #BIT7+BIT5+BIT4+BIT2+BIT1,R3
4052  020066  004237  020100                  JSR     R2,10$          ;S= RAM OUTPUT. D=DVRIC
4053  020072  000204                          BIT7+BIT2
4054  020074  000200                          BIT7
4055  020076  104400                          SCOPE
4056  020100  012710  050000          10$:    MOV     #S.C,(R0) ;SET CLEAR INSTR
4057  020104  010201                          MOV     R2,R1           ;JSR PC TO R1
4058  020106  052210                          BIS     (R2)+,(R0)      ;LOAD SET/CLEAR POINT
4059  020110  104415                          ROMCLK                  ;EXECUTE
4060  020112  010310                          MOV     R3,(R0)         ;LOAD XFER
4061  020114  104415                          ROMCLK                  ;EXECUTE
4062  020116  017704  161244                  MOV     @DVRIC,R4       ;READ RESULTS
4063  020122  020504                          CMP     R5,R4           ;OK??
4064  020124  001401                          BEQ     11$             ;YES
4065  020126  104005                          HLT     5               ;XFER FAILED
4066  020130  012710  050000          11$:    MOV     #S.C,(R0) ;SET/CLEAR
4067  020134  052210                          BIS     (R2)+,(R0)      ;POINT
4068  020136  104415                          ROMCLK                  ;EXECUTE
4069  020140  010310                          MOV     R3,(R0)         ;XFER
4070  020142  104415                          ROMCLK                  ;EXECUTE
4071  020144  005005                          CLR     R5              ;SET EXPECTED RESULTS
4072  020146  017704  161214                  MOV     @DVRIC,R4       ;READ REAL RESULTS
4073  020152  001401                          BEQ     12$             ;BR IF RESULT=0
4074  020154  104005                          HLT     5               ;DVRICR NOT =0
4075  020156  000202          12$:    RTS     R2              ;EXIT SUB
4076
4077
4078                                          ;************************* TEST 74 *********************************
4079                                          ;*BASIC TEST OF THE 'ALU OPERATION' INSTRUCTION.
4080                                          ;*FIRST PART:ISSUE AN INIT AND MOVE
4081                                          ;*THE ALU RESULT REGISTER TO THE DVRIC
4082                                          ;*REGISTER VERIFING THAT IT IS ZERO.
4083                                          ;*SECOND PART: DO A FUNCTION 'F=A'
4084                                          ;*THEN MOV 'F' TO THE DVRIC REGISTER VERIFING IT TO
4085                                          ;*BE ZERO
4086                                          ;*THIRD PART: DO A FUNCTION 'F=A+B'; MOVING
4087                                          ;*'F' TO DVRIC AND MAKING SURE IT IS ZERO.
4088                                          ;*THUS THE FOLLOWING HAS BEEN TESTED:
4089                                          ;*ALU RESULT,'A' REG,AND 'B' REG ALL ZEROED ON INIT.
4090                                          ;;*****************************************************************
4091
4092                                  ;  TEST 74
4093                                  ;--------------
```

```
4094  020160  012737  000074  001226  TST74:  MOV     #74,TSTNO
4095  020166  012737  020300  001216          MOV     #TST75,NEXT
4096  020174  104412                          MSTCLR                  ;RESET DV11
4097  020176  012777  000010  161156          MOV     #BIT3,@DVSCR    ;SET SOURCE SELECT
4098  020204  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER IN R0
4099  020210  012702  030366                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT2+BIT1,R2
4100  020214  010210                          MOV     R2,(R0)         ;XFR ''ALU RESULT REG.'' TO DVNSR
4101  020216  104415                          ROMCLK                  ;CLOCK INSTRUCTION
4102  020220  005005                          CLR     R5              ;ZERO ''EXPECTED'' LOC
4103  020222  017704  161140                  MOV     @DVRIC,R4       ;READ ''ALU RESULT''
4104  020226  001401                          BEQ     1$              ;S/B=0
4105  020230  104006                          HLT     6               ;ALU RESULT NOT=0 ON INIT
4106  020232  012710  010037          1$:     MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,(R0)
4107  020236  104415                          ROMCLK                  ;DO ''ALU F=A''
4108  020240  010210                          MOV     R2,(R0)         ;XFR ''ALU RESULT'' TO DVNSR
4109  020242  104415                          ROMCLK                  ;CLOCK INSTR
4110  020244  017704  161116                  MOV     @DVRIC,R4       ;READ RESULT
4111  020250  001401                          BEQ     2$              ;S/B=0
4112  020252  104006                          HLT     6               ;''A'' REG NOT=0 ON INIT
4113  020254  012710  010026          2$:     MOV     #ALU+BIT4+BIT2+BIT1,(R0)
4114  020260  104415                          ROMCLK                  ;ALU F=A+B
4115  020262  010210                          MOV     R2,(R0)         ;XFR TO RIC
4116  020264  104415                          ROMCLK                  ;CLOCK INSTR.
4117  020266  017704  161074                  MOV     @DVRIC,R4       ;READ RESULT
4118  020272  001401                          BEQ     3$              ;S/B=0
4119  020274  104006                          HLT     6               ;''B'' REG NOT=0 ON INIT
4120  020276  104400          3$:     SCOPE                           ;SCOPE TEST
4121
4122                                  ;************************** TEST 75 ********************************
4123                                  ;*TEST OF ALU OPERATIONS.
4124                                  ;*TEST OF ALL ALU OPERATIONS USED BY DV11.
4125                                  ;*FUNCTIONS TESTED:(NOTE THAT ''F'' IS ALU RESULT)
4126                                  ;*DVSFR BITS:
4127                                  ;*BIT12 BIT05 BIT04 BIT03 BIT02 BIT01 BIT00 FUNCTION
4128                                  ;*  1     0     1     1     1     0     0   F=-1
4129                                  ;*  1     0     0     1     1     0     0   F=0
4130                                  ;*  1     0     1     1     1     1     1   F=A
4131                                  ;*  1     0     0     0     1     0     1   F=B
4132                                  ;*  1     1     1     1     1     1     1   F=A+1
4133                                  ;*  1     0     1     0     1     1     0   F=A+B
4134                                  ;;*******************************************************************
4135
4136                          ;  TEST 75
4137                          ;--------------
4138  020300  012737  000075  001226  TST75:  MOV     #75,TSTNO
4139  020306  012737  021210  001216          MOV     #TST76,NEXT
4140                                  ;*FUNCTION TESTED
4141                                  ;*F=-1,RIC_F
4142                                  ;*
4143  020314  012737  020322  001220          MOV     #10$,LOCK       ;SET FOR SW09
4144  020322  104412          10$:    MSTCLR                          ;CLEAR DV11
4145  020324  012777  000010  161030          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
4146  020332  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER IN R0
4147  020336  012702  030366                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT2+BIT1,R2
4148  020342  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
4149  020346  104415                          ROMCLK                  ;DO ''F=-1''
```

```
4150  020350  010210                          MOV     R2,(R0)         ;XFR "ALU RESULT TO DVRIC"
4151  020352  104415                          ROMCLK                  ;
4152  020354  017704  161006                  MOV     @DVRIC,R4       ;READ RESULTS
4153  020360  012705  177777                  MOV     #-1,R5          ;SET EXPECTED
4154  020364  020504                          CMP     R5,R4           ;DID F=-1 WORK?
4155  020366  001401                          BEQ     63$             ;BR IF YES
4156  020370  104006                          HLT     6               ;F=-1 APPEARED TO FAIL
4157                                          ;*TEST THAT DVRIC (NOW THAT ITS ALL 1'S)
4158                                          ;*CAN BE CLEARED BY A MSTCLR.
4159                                          ;*
4160  020372  104401                63$:      SCOP1                   ;SW09=1?
4161  020374  012737  020402  001220          MOV     #11$,LOCK       ;SET RETURN IF SW09=1
4162  020402  104412                11$:      MSTCLR                  ;CLEAR DV11
4163  020404  005005                          CLR     R5              ;SET EXPECTED RESULTS
4164  020406  017704  160754                  MOV     @DVRIC,R4       ;READ DVRIC
4165  020412  001401                          BEQ     64$             ;BR IF=0
4166  020414  104006                          HLT     6               ;DVRIC REG. NOT CLEARED ON INIT
4167                                          ;*NEXT SET OF FUNCTIONS:
4168                                          ;*F=0,RIC_F
4169                                          ;*
4170  020416  012777  000010  160736  64$:    MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
4171  020424  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
4172  020430  104415                          ROMCLK                  ;"F=-1"
4173  020432  012710  010014                  MOV     #ALU+BIT3+BIT2,(R0)
4174  020436  104415                          ROMCLK                  ;"F=0"
4175  020440  010210                          MOV     R2,(R0)         ;XFR "F" TO DVRIC
4176  020442  104415                          ROMCLK                  ;
4177  020444  005005                          CLR     R5              ;SET EXPECTED
4178  020446  017704  160714                  MOV     @DVRIC,R4       ;READ RESULTS
4179  020452  001401                          BEQ     65$             ;BR IF=0
4180  020454  104006                          HLT     6               ;"F=0" APPEARED TO FAIL
4181                                          ;*NEXT SET OF FUNCTIONS:
4182                                          ;*F=A+1,RIC_F,A_F,F=0,F=A
4183                                          ;*
4184  020456  104401                65$:      SCOP1                   ;SW09=1?
4185  020460  012737  020466  001220          MOV     #12$,LOCK       ;SET RETURN
4186  020466  012705  000001        12$:      MOV     #1,R5           ;SET GOOD RESULTS
4187  020472  012710  010077        1$:       MOV     #ALU+BIT5+BIT4+BIT3+BIT2+BIT1+BIT0,(R0)
4188  020476  052777  000002  160656          BIS     #BIT1,@DVSCR    ;ISSUE ROM CLOCK
4189  020504  010210                          MOV     R2,(R0)         ;XFR "F" TO DVRIC
4190  020506  052777  000002  160646          BIS     #BIT1,@DVSCR    ;ISSUE ROM CLK
4191  020514  017704  160646                  MOV     @DVRIC,R4       ;READ RESULTS
4192  020520  020504                          CMP     R5,R4           ;FUNCTION WORK?
4193  020522  001401                          BEQ     66$             ;YES
4194  020524  104006                          HLT     6               ;F=A+1 APPEARED TO FAIL
4195  020526  012710  030361        66$:      MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,(R0)
4196  020532  052777  000002  160622          BIS     #BIT1,@DVSCR    ;ISSUE ROM CLK
4197  020540  012710  010014                  MOV     #ALU+BIT3+BIT2,(R0)
4198  020544  052777  000002  160610          BIS     #BIT1,@DVSCR    ;ROM CLK
4199  020552  010210                          MOV     R2,(R0)         ;XFR RIC_F
4200  020554  052777  000002  160600          BIS     #BIT1,@DVSCR    ;ROM CLK
4201  020562  017704  160600                  MOV     @DVRIC,R4       ;READ RESULT
4202  020566  001401                          BEQ     67$             ;BR IF GOOD
4203  020570  104000                          HLT     0               ;F=0 FAILED
4204  020572  012710  010037        67$:      MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,(R0)
4205  020576  052777  000002  160556          BIS     #BIT1,@DVSCR    ;CLOCK "F=A"
```

```
4206  020604  010210                          MOV     R2,(R0)          ;XFR RIC_F
4207  020606  052777  000002  160546          BIS     #BIT1,@DVSCR     ;ROM CLK
4208  020614  017704  160546                  MOV     @DVRIC,R4        ;READ RESULT
4209  020620  020504                          CMP     R5,R4            ;BR IF OK
4210  020622  001401                          BEQ     68$              ;
4211  020624  104006                          HLT     6                ;F=A FAILED
4212  020626  005205                  68$:    INC     R5               ;UPDATE DATA
4213  020630  001320                          BNE     1$               ;BR IF NOT ALL DONE.
4214  020632  104401                          SCOP1                    ;SW09=1?
4215                                          ;*NEXT SET OF FUNCTIONS:
4216                                          ;*F=A+1,A_F,B_F,F=A+B,RIC_F
4217                                          ;*
4218  020634  012737  020642  001220          MOV     #13$,LOCK        ;SET RETURN
4219  020642  104412                  13$:    MSTCLR                   ;RESET DV11
4220  020644  012777  000010  160510          MOV     #BIT3,@DVSCR     ;SET SOURCE SEL.
4221  020652  012705  000002                  MOV     #2,R5            ;SET EXPECTED RESULTS
4222  020656  013701  001366                  MOV     DVRIC,R1         ;SET POINTER
4223  020662  013703  001362                  MOV     DVSCR,R3         ;SET POINTER
4224  020666  012710  010077                  MOV     #ALU+BIT5+BIT4+BIT3+BIT2+BIT1+BIT0,(R0)
4225  020672  104415                          ROMCLK                   ;F=A+1
4226  020674  012710  030361                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,(R0)
4227  020700  104415                          ROMCLK                   ;XFR A_ALU RESULT
4228  020702  012710  030362          2$:     MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
4229  020706  052713  000002                  BIS     #BIT1,(R3)       ;XFR B_ALU RESULT
4230  020712  012710  010026                  MOV     #ALU+BIT4+BIT2+BIT1,(R0)
4231  020716  052713  000002                  BIS     #BIT1,(R3)       ;CLOCK F=A+B
4232  020722  010210                          MOV     R2,(R0)          ;XFR RIC_F
4233  020724  052713  000002                  BIS     #BIT1,(R3)       ;
4234  020730  011104                          MOV     (R1),R4          ;READ DVRIC
4235  020732  020504                          CMP     R5,R4            ;FUNCTION WORK?
4236  020734  001401                          BEQ     69$              ;BR IF YES
4237  020736  104006                          HLT     6                ;F=A+B APPEARED TO FAIL
4238  020740  005205                  69$:    INC     R5               ;INC DATA POINTER
4239  020742  022705  000002                  CMP     #2,R5            ;ALL DONE?
4240  020746  001355                          BNE     2$               ;BR IF NO
4241  020750  104412                          MSTCLR                   ;RESET DV11
4242  020752  017704  160410                  MOV     @DVRIC,R4        ;DVRIC ZERO ON INIT?
4243  020756  001401                          BEQ     70$              ;BR IF YES
4244  020760  104000                          HLT     0                ;S/B=0
4245  020762  104401                  70$:    SCOP1                    ;SW09=1?
4246                                          ;*NEXT SET OF FUNCTIONS:
4247                                          ;*F=-1,B_F,F=0,F=B
4248                                          ;*
4249  020764  012737  020772  001220          MOV     #14$,LOCK        ;SET RETURN
4250  020772  104412                  14$:    MSTCLR                   ;RESET DV11
4251  020774  012777  000010  160360          MOV     #BIT3,@DVSCR     ;SET SOURCE SEL.
4252  021002  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
4253  021006  104415                          ROMCLK                   ;F=-1
4254  021010  012710  030362                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
4255  021014  104415                          ROMCLK                   ;XFR B_F
4256  021016  012710  010014                  MOV     #ALU+BIT3+BIT2,(R0) ;
4257  021022  104415                          ROMCLK                   ;F=0
4258  021024  012710  010005                  MOV     #ALU+BIT2+BIT0,(R0) ;
4259  021030  104415                          ROMCLK                   ;F=B
4260  021032  010210                          MOV     R2,(R0)          ;XFR RIC_F
4261  021034  104415                          ROMCLK                   ;CLOCK
```

```
4262  021036  017704  160324                  MOV     @DVRIC,R4       ;READ RESULTS
4263  021042  012705  177777                  MOV     #-1,R5          ;SET EXPECTED
4264  021046  020504                          CMP     R5,R4           ;DID F=B WORK?
4265  021050  001401                          BEQ     71$             ;BR IF YES
4266  021052  104006                          HLT     6               ;F=B FAILED
4267  021054  104401                  71$:    SCOP1                   ;SW09=1?
4268                                          ;*NEXT SET OF FUNCTIONS:
4269                                          ;*CATCH "SET/CLEAR" THAT WAS MISSED.
4270                                          ;*F=-1,S/C[ALU01=0],RIC_F
4271                                          ;*
4272  021056  012737  021064  001220          MOV     #15$,LOCK       ;SET RETURN
4273  021064  104412                  15$:    MSTCLR                  ;RESET DV11
4274  021066  012777  000010  160266          MOV     #BIT3,@DVSCR    ;SET SOURCE SELECT
4275  021074  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
4276  021100  104415                          ROMCLK                  ;F=-1
4277  021102  012710  050026                  MOV     #S.C+BIT4+BIT2+BIT1,(R0)
4278  021106  104415                          ROMCLK                  ;S/C "CLEAR ALU01"
4279  021110  010210                          MOV     R2,(R0)         ;XFR RIC_F
4280  021112  104415                          ROMCLK                  ;
4281  021114  011104                          MOV     (R1),R4         ;READ DVRIC
4282  021116  012705  177775                  MOV     #177775,R5      ;SET EXPECTED
4283  021122  020504                          CMP     R5,R4           ;DID S/C ALU01 WORK?
4284  021124  001401                          BEQ     72$             ;BR IF YES
4285  021126  104006                          HLT     6               ;S/C ALU01 FAILED.
4286  021130  104401                  72$:    SCOP1                   ;SW09=1?
4287                                          ;*NEXT SET OF FUNCTIONS:
4288                                          ;*CATCH ANOTHER "SET/CLEAR" THAT WAS MISSED.
4289                                          ;*F=-1,S/C[ALU HIGH BYTE=0],RIC_F
4290                                          ;*
4291  021132  012737  021140  001220          MOV     #16$,LOCK       ;SET RETURN
4292  021140  104412                  16$:    MSTCLR                  ;RESET DV11
4293  021142  012777  000010  160212          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
4294  021150  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
4295  021154  104415                          ROMCLK                  ;F=-1
4296  021156  012710  050027                  MOV     #S.C+BIT4+BIT2+BIT1+BIT0,(R0)
4297  021162  104415                          ROMCLK                  ;S/C "CLEAR ALU HIGH BYTE"
4298  021164  010210                          MOV     R2,(R0)         ;XFR RIC_F
4299  021166  104415                          ROMCLK                  ;
4300  021170  011104                          MOV     (R1),R4         ;READ RIC
4301  021172  012705  000377                  MOV     #377,R5         ;SET EXPECTED
4302  021176  020504                          CMP     R5,R4           ;DID S/C WORK?
4303  021200  001401                          BEQ     73$             ;BR IF YES
4304  021202  104006                          HLT     6               ;S/C ALU HIGH BYTE FAILED.
4305  021204  104401                  73$:    SCOP1                   ;SW09=1?
4306  021206  104400                          SCOPE                   ;SCOPE TEST
4307
4308
4309                                          ;************************* TEST 76 ******************************
4310                                          ;*MASTER SCANNER TEST.
4311                                          ;*VERIFY FIRST THAT THE MASTER SCANNER
4312                                          ;*IS CLEARED BY INIT.
4313                                          ;*VERIFY SECONDLY THAT THE MASTER SCANNER
4314                                          ;*CAN BE INCREMENTED FROM 0 THRU 17 BACK TO 0.
4315                                          ;;*****************************************************************
4316
4317                                  ;  TEST 76
```

```
                                    ;--------------
001226  TST76:  MOV     #76,TSTNO
001216           MOV     #TST77,NEXT
                 MSTCLR                      ;RESET DV11
160126           MOV     #BIT3,@DVSCR        ;SET SOURCE SEL
001400           MOV     DVSFR,R0            ;SET DVSFR POINTER
030126           MOV     #XFR+BIT6+BIT4+BIT2+BIT1,R2
                 MOV     R2,(R0)             ;XFR 'MASTER SCAN 0-3'' TO DVRIC
                 ROMCLK                      ;
                 CLR     R5                  ;SET EXPECTED
160110           MOV     @DVRIC,R4           ;READ RESULTS
                 BEQ     1$                  ;BR IF=0
                 HLT     6                   ;MSCAN NOT=0 ON INIT
                1$:      INC     R5          ;UPDATE POINTER
000025           MOV     #25,R3              ;SET COUNT TO 25
050102          2$:      MOV     #S.C+BIT6+BIT1,(R0) ;S/C ''ADVANCE MSCAN''
                 ROMCLK                      ;CLOCK
050102           MOV     #S.C+BIT6+BIT1,(R0) ;S/C ''ADVANCE MSCAN''
                 ROMCLK                      ;CLOCK
                 MOV     R2,(R0)             ;XFR RIC_MSCAN
                 ROMCLK                      ;CLOCK
160052           MOV     @DVRIC,R4           ;READ RESULTS
                 CMP     R5,R4               ;MSCAN INCREMENTED?
                 BEQ     3$                  ;BR IF YES
                 HLT     6                   ;MSCAN WRONG
                3$:      INC     R5          ;UPDATE
177760           BIC     #^C<17>,R5          ;CLEAN
                 DEC     R3                  ;COUNT DONE?
                 BNE     2$                  ;BR IF NO
                 SCOPE                       ;SCOPE

                 ;************************ TEST 77 ******************************
                 ;*BASIC TESTS OF THE 'RAM OPERATION'' INSTRUCTION.
                 ;*VERIFY THE READ PORTION OF THE RAM OPERATION.
                 ;*LOAD ALL SECONDARY REGISTERS OF ALL LINES
                 ;*WITH DIFFERENT NUMBERS AND VERIFY THAT THE RAM OPERATION
                 ;*CAN READ THE CORRECT SEC. REG. INTO THE DVRIC REG.
                 ;:*************************************************************

                 ;  TEST 77
                 ;--------------
001226  TST77:  MOV     #77,TSTNO
001216           MOV     #TST100,NEXT
                 MSTCLR                      ;RESET DV11
                 CLR     R0                  ;
                 CLR     R1                  ;
001372           MOV     DVSRS,R2            ;
001374           MOV     DVSRSH,R3           ;
001376           MOV     DVSRA,R5            ;
                 MOV     #1,R4               ;
                1$:      MOVB    R0,(R2)     ;LOAD LINE NUMBER
                 MOVB    R1,(R3)             ;LOAD SEC. REG. POINTER
                 MOV     R4,(R5)             ;LOAD DATA
                 CMPB    (R0)+,(R4)+         ;UPDATE LINE AND DATA
000020           CMP     #16.,R0             ;ALL LINES DONE?
                 BNE     1$                  ;BR IF NO
```

```
4374  021416  005000                          CLR     R0              ;ZERO LINE POINTER
4375  021420  005201                          INC     R1              ;UPDATE SEC REG POINTER.
4376  021422  022701  000020                  CMP     #16.,R1         ;ALL SEC REG POINTERS DONE?
4377  021426  001364                          BNE     1$              ;BR IF NO
4378  021430  005077  157736          2$:     CLR     @DVSRS          ;ZERO POINTERS
4379  021434  012705  000001                  MOV     #1,R5           ;SET GOOD DATA
4380  021440  012777  000010  157714          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
4381  021446  012703  020000                  MOV     #RAM,R3         ;LOAD RAM INSTR.
4382  021452  013700  001400                  MOV     DVSFR,R0        ;SET POINTER
4383  021456  012702  030266                  MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT1,R2
4384  021462  010310                  3$:     MOV     R3,(R0)         ;DO RAM READ
4385  021464  104415                          ROMCLK                  ;EXECUTE
4386  021466  010210                          MOV     R2,(R0)         ;XFR RIC_RAM OUTPUT
4387  021470  104415                          ROMCLK                  ;CLOCK
4388  021472  017704  157670                  MOV     @DVRIC,R4       ;READ RESULT
4389  021476  020504                          CMP     R5,R4           ;GOOD?
4390  021500  001401                          BEQ     4$              ;BR IF YES
4391  021502  104006                          HLT     6               ;RAM READ FAILED.
4392  021504  062705  000020          4$:     ADD     #20,R5          ;UPDATE DATA
4393  021510  005203                          INC     R3              ;UPDATE POINTER
4394  021512  122703  000020                  CMPB    #16.,R3         ;ALL DONE
4395  021516  001361                          BNE     3$              ;BR IF NO
4396  021520  042705  177760                  BIC     #^C<17>,R5      ;CLEAR JUNK
4397  021524  012703  020000                  MOV     #RAM,R3         ;SET RAM INSTR
4398  021530  010046                          MOV     R0,-(SP)        ;SAVE R0 ON STACK
4399  021532  004537  031676                  PERFORM ,SETSCAN         ;UPDATE MSCANNER
4400  021536  000001                          1                       ;
4401  021540  012600                          MOV     (SP)+,R0        ;RESTORE R0
4402  021542  005205                          INC     R5              ;UPDATE DATA
4403  021544  012710  030124                  MOV     #XFR+BIT6+BIT4+BIT2,(R0) ;XFR RAR_MSCAN 0-3
4404  021550  104415                          ROMCLK                  ;EXECUTE
4405  021552  022705  000020                  CMP     #16.,R5         ;ALL DONE?
4406  021556  001341                          BNE     3$              ;BR IF NO
4407  021560  104400                          SCOPE                   ;SCOPE TEST.
4408
4409                                  ;********************** TEST 100 ******************************
4410                                  ;*TEST OF BRANCH A TEST POINTS
4411                                  ;*THAT WERE PREVIOUSLY SKIPPED BECAUSE
4412                                  ;OF THE SIGNALS NEEDED TO TEST THE
4413                                  ;*FUNCTIONS:
4414                                  ;*BIT11 BIT10   BIT09   BIT08   FUNCTION
4415                                  ;*  0     0       0       0     ALU 15=1,0
4416                                  ;*  1     0       0       0     ALU 13-  =1,0
4417                                  ;*  1     0       0       0           -12=1,0
4418                                  ;*  1     0       1       0     ALU 00=1,0
4419                                  ;*  1     0       1       1     ALU 01=1,0
4420                                  ;*  1     1       0       0     ALU 02=1,0
4421                                  ;*  1     1       0       1     ALU 03=1,0
4422                                  ;*  1     1       1       0     ALU 04=1,0
4423                                  ;:*************************************************************
4424
4425                          ;  TEST 100
4426                          ;--------------
4427  021562  012737  000100  001226  TST100: MOV     #100,TSTNO
4428  021570  012737  023164  001216          MOV     #TST101,NEXT
4429  021576  005077  157570                  CLR     @DVSRS
```

```
4430                                        ;*BRANCH "A" TEST OF ALU 15
4431                                        ;*
4432  021602  104412                        MSTCLR                          ;RESET DV11
4433  021604  012777  000010  157550        MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
4434  021612  012777  100000  157556        MOV     #BIT15,@DVSRA           ;LOAD DATA
4435  021620  012777  020000  157552        MOV     #RAM,@DVSFR             ;DO A "RAM READ"
4436  021626  104415                        ROMCLK                          ;EXECUTE
4437  021630  012777  030261  157542        MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
4438  021636  104415                        ROMCLK                          ;XFR RAM OUTPUT TO "A" REG
4439  021640  012777  010037  157532        MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
4440  021646  104415                        ROMCLK                          ;F=A
4441  021650  012777  000000  157522        MOV     #0000,@DVSFR            ;LOAD BRANCH POINT
4442  021656  017704  157506                MOV     @DVLCR,R4               ;READ BRANCH TEST POINT
4443  021662  042704  177774                BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
4444  021666  012705  000002                MOV     #BIT1,R5                ;SET EXPECTED
4445  021672  020504                        CMP     R5,R4                   ;BR POINTS CORRECT?
4446  021674  001401                        BEQ     64$                     ;BR IF YES
4447  021676  104006                        HLT     6                       ;BRANCH POINTS WRONG
4448  021700  104412              64$:      MSTCLR
4449  021702  012777  000010  157452        MOV     #BIT3,@DVSCR            ;SET SOURCE SEL.
4450  021710  012777  000000  157462        MOV     #0000,@DVSFR            ;RESET DV11
4451  021716  017704  157446                MOV     @DVLCR,R4               ;LOAD BRANCH. POINT TEST
4452  021722  042704  177774                BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
4453  021726  012705  000003                MOV     #BIT1+BIT0,R5           ;CLEAR JUNK
4454  021732  020504                        CMP     R5,R4                   ;SET EXPECTED
4455  021734  001401                        BEQ     65$                     ;BR POINT OK?
4456  021736  104006                        HLT     6                       ;BR IF YES
4457  021740                        65$:                                      ;BR POINTS WRONG
4458                                        ;*BRANCH "A" TEST OF ALU 13-
4459                                        ;*
4460  021740  104412                        MSTCLR                          ;RESET DV11
4461  021742  012777  000010  157412        MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
4462  021750  012777  020000  157420        MOV     #BIT13,@DVSRA           ;LOAD DATA
4463  021756  012777  020000  157414        MOV     #RAM,@DVSFR             ;DO A "RAM READ"
4464  021764  104415                        ROMCLK                          ;EXECUTE
4465  021766  012777  030261  157404        MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
4466  021774  104415                        ROMCLK                          ;XFR RAM OUTPUT TO "A" REG
4467  021776  012777  010037  157374        MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
4468  022004  104415                        ROMCLK                          ;F=A
4469  022006  012777  004000  157364        MOV     #BIT11,@DVSFR           ;LOAD BRANCH POINT
4470  022014  017704  157350                MOV     @DVLCR,R4               ;READ BRANCH TEST POINT
4471  022020  042704  177774                BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
4472  022024  012705  000002                MOV     #BIT1,R5                ;SET EXPECTED
4473  022030  020504                        CMP     R5,R4                   ;BR POINTS CORRECT?
4474  022032  001401                        BEQ     66$                     ;BR IF YES
4475  022034  104006                        HLT     6                       ;BRANCH POINTS WRONG
4476  022036  104412              66$:      MSTCLR
4477  022040  012777  000010  157314        MOV     #BIT3,@DVSCR            ;SET SOURCE SEL.
4478  022046  012777  004000  157324        MOV     #BIT11,@DVSFR           ;RESET DV11
4479  022054  017704  157310                MOV     @DVLCR,R4               ;LOAD BRANCH. POINT TEST
4480  022060  042704  177774                BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
4481  022064  012705  000003                MOV     #BIT1+BIT0,R5           ;CLEAR JUNK
4482  022070  020504                        CMP     R5,R4                   ;SET EXPECTED
4483  022072  001401                        BEQ     67$                     ;BR POINT OK?
4484  022074  104006                        HLT     6                       ;BR IF YES
4485  022076                        67$:                                      ;BR POINTS WRONG
```

```
4486                                            ;*BRANCH "A" TEST OF ALU   -12
4487                                            ;*
4488  022076  104412                           MSTCLR                          ;RESET DV11
4489  022100  012777  000010  157254           MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
4490  022106  012777  010000  157262           MOV     #BIT12,@DVSRA           ;LOAD DATA
4491  022114  012777  020000  157256           MOV     #RAM,@DVSFR             ;DO A "RAM READ"
4492  022122  104415                           ROMCLK                          ;EXECUTE
4493  022124  012777  030261  157246           MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
4494  022132  104415                           ROMCLK                          ;XFR RAM OUTPUT TO "A" REG
4495  022134  012777  010037  157236           MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
4496  022142  104415                           ROMCLK                          ;F=A
4497  022144  012777  004000  157226           MOV     #BIT11,@DVSFR           ;LOAD BRANCH POINT
4498  022152  017704  157212                   MOV     @DVLCR,R4               ;READ BRANCH TEST POINT
4499  022156  042704  177774                   BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
4500  022162  012705  000002                   MOV     #BIT1,R5                ;SET EXPECTED
4501  022166  020504                           CMP     R5,R4                   ;BR POINTS CORRECT?
4502  022170  001401                           BEQ     68$                     ;BR IF YES
4503  022172  104006                           HLT     6                       ;BRANCH POINTS WRONG
4504  022174  104412                   68$:    MSTCLR
4505  022176  012777  000010  157156           MOV     #BIT3,@DVSCR            ;SET SOURCE SEL.
4506  022204  012777  004000  157166           MOV     #BIT11,@DVSFR           ;RESET DV11
4507  022212  017704  157152                   MOV     @DVLCR,R4               ;LOAD BRANCH. POINT TEST
4508  022216  042704  177774                   BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
4509  022222  012705  000003                   MOV     #BIT1+BIT0,R5           ;CLEAR JUNK
4510  022226  020504                           CMP     R5,R4                   ;SET EXPECTED
4511  022230  001401                           BEQ     69$                     ;BR POINT OK?
4512  022232  104006                           HLT     6                       ;BR IF YES
4513  022234                           69$:                                    ;BR POINTS WRONG
4514                                            ;*BRANCH "A" TEST OF ALU 00
4515                                            ;*
4516  022234  104412                           MSTCLR                          ;RESET DV11
4517  022236  012777  000010  157116           MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
4518  022244  012777  000001  157124           MOV     #BIT0,@DVSRA            ;LOAD DATA
4519  022252  012777  020000  157120           MOV     #RAM,@DVSFR             ;DO A "RAM READ"
4520  022260  104415                           ROMCLK                          ;EXECUTE
4521  022262  012777  030261  157110           MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
4522  022270  104415                           ROMCLK                          ;XFR RAM OUTPUT TO "A" REG
4523  022272  012777  010037  157100           MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
4524  022300  104415                           ROMCLK                          ;F=A
4525  022302  012777  005000  157070           MOV     #BIT11+BIT9,@DVSFR      ;LOAD BRANCH POINT
4526  022310  017704  157054                   MOV     @DVLCR,R4               ;READ BRANCH TEST POINT
4527  022314  042704  177774                   BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
4528  022320  012705  000002                   MOV     #BIT1,R5                ;SET EXPECTED
4529  022324  020504                           CMP     R5,R4                   ;BR POINTS CORRECT?
4530  022326  001401                           BEQ     70$                     ;BR IF YES
4531  022330  104006                           HLT     6                       ;BRANCH POINTS WRONG
4532  022332  104412                   70$:    MSTCLR
4533  022334  012777  000010  157020           MOV     #BIT3,@DVSCR            ;SET SOURCE SEL.
4534  022342  012777  005000  157030           MOV     #BIT11+BIT9,@DVSFR      ;RESET DV11
4535  022350  017704  157014                   MOV     @DVLCR,R4               ;LOAD BRANCH. POINT TEST
4536  022354  042704  177774                   BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
4537  022360  012705  000003                   MOV     #BIT1+BIT0,R5           ;CLEAR JUNK
4538  022364  020504                           CMP     R5,R4                   ;SET EXPECTED
4539  022366  001401                           BEQ     71$                     ;BR POINT OK?
4540  022370  104006                           HLT     6                       ;BR IF YES
4541  022372                           71$:                                    ;BR POINTS WRONG
```

```
4542                                            ;*BRANCH "A" TEST OF ALU 01
4543                                            ;*
4544  022372  104412                            MSTCLR                          ;RESET DV11
4545  022374  012777  000010  156760            MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
4546  022402  012777  000002  156766            MOV     #BIT1,@DVSRA            ;LOAD DATA
4547  022410  012777  020000  156762            MOV     #RAM,@DVSFR             ;DO A "RAM READ"
4548  022416  104415                            ROMCLK                          ;EXECUTE
4549  022420  012777  030261  156752            MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
4550  022426  104415                            ROMCLK                          ;XFR RAM OUTPUT TO "A" REG
4551  022430  012777  010037  156742            MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
4552  022436  104415                            ROMCLK                          ;F=A
4553  022440  012777  005400  156732            MOV     #BIT11+BIT9+BIT8,@DVSFR ;LOAD BRANCH POINT
4554  022446  017704  156716                    MOV     @DVLCR,R4               ;READ BRANCH TEST POINT
4555  022452  042704  177774                    BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
4556  022456  012705  000002                    MOV     #BIT1,R5                ;SET EXPECTED
4557  022462  020504                            CMP     R5,R4                   ;BR POINTS CORRECT?
4558  022464  001401                            BEQ     72$                     ;BR IF YES
4559  022466  104006                            HLT     6                       ;BRANCH POINTS WRONG
4560  022470  104412                    72$:    MSTCLR
4561  022472  012777  000010  156662            MOV     #BIT3,@DVSCR            ;SET SOURCE SEL.
4562  022500  012777  005400  156672            MOV     #BIT11+BIT9+BIT8,@DVSFR ;RESET DV11
4563  022506  017704  156650                    MOV     @DVLCR,R4               ;LOAD BRANCH. POINT TEST
4564  022512  042704  177774                    BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
4565  022516  012705  000003                    MOV     #BIT1+BIT0,R5           ;CLEAR JUNK
4566  022522  020504                            CMP     R5,R4                   ;SET EXPECTED
4567  022524  001401                            BEQ     73$                     ;BR POINT OK?
4568  022526  104006                            HLT     6                       ;BR IF YES
4569  022530                            73$:                                  ;BR POINTS WRONG
4570                                            ;*BRANCH "A" TEST OF ALU 02
4571                                            ;*
4572  022530  104412                            MSTCLR                          ;RESET DV11
4573  022532  012777  000010  156622            MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
4574  022540  012777  000004  156630            MOV     #BIT2,@DVSRA            ;LOAD DATA
4575  022546  012777  020000  156624            MOV     #RAM,@DVSFR             ;DO A "RAM READ"
4576  022554  104415                            ROMCLK                          ;EXECUTE
4577  022556  012777  030261  156614            MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
4578  022564  104415                            ROMCLK                          ;XFR RAM OUTPUT TO "A" REG
4579  022566  012777  010037  156604            MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
4580  022574  104415                            ROMCLK                          ;F=A
4581  022576  012777  006000  156574            MOV     #BIT11+BIT10,@DVSFR     ;LOAD BRANCH POINT
4582  022604  017704  156560                    MOV     @DVLCR,R4               ;READ BRANCH TEST POINT
4583  022610  042704  177774                    BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
4584  022614  012705  000002                    MOV     #BIT1,R5                ;SET EXPECTED
4585  022620  020504                            CMP     R5,R4                   ;BR POINTS CORRECT?
4586  022622  001401                            BEQ     74$                     ;BR IF YES
4587  022624  104006                            HLT     6                       ;BRANCH POINTS WRONG
4588  022626  104412                    74$:    MSTCLR
4589  022630  012777  000010  156524            MOV     #BIT3,@DVSCR            ;SET SOURCE SEL.
4590  022636  012777  006000  156534            MOV     #BIT11+BIT10,@DVSFR     ;RESET DV11
4591  022644  017704  156520                    MOV     @DVLCR,R4               ;LOAD BRANCH. POINT TEST
4592  022650  042704  177774                    BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
4593  022654  012705  000003                    MOV     #BIT1+BIT0,R5           ;CLEAR JUNK
4594  022660  020504                            CMP     R5,R4                   ;SET EXPECTED
4595  022662  001401                            BEQ     75$                     ;BR POINT OK?
4596  022664  104006                            HLT     6                       ;BR IF YES
4597  022666                            75$:                                  ;BR POINTS WRONG
```

```
                                                ;*BRANCH "A" TEST OF ALU 03
                                                ;*
022666  104412                                  MSTCLR                          ;RESET DV11
022670  012777  000010  156464                  MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
022676  012777  000010  156472                  MOV     #BIT3,@DVSRA    ;LOAD DATA
022704  012777  020000  156466                  MOV     #RAM,@DVSFR     ;DO A "RAM READ"
022712  104415                                  ROMCLK                          ;EXECUTE
022714  012777  030261  156456                  MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
022722  104415                                  ROMCLK                          ;XFR RAM OUTPUT TO "A" REG
022724  012777  010037  156446                  MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
022732  104415                                  ROMCLK                          ;F=A
022734  012777  006400  156436                  MOV     #BIT11+BIT10+BIT8,@DVSFR        ;LOAD BRANCH POINT
022742  017704  156422                          MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
022746  042704  177774                          BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
022752  012705  000002                          MOV     #BIT1,R5        ;SET EXPECTED
022756  020504                                  CMP     R5,R4           ;BR POINTS CORRECT?
022760  001401                                  BEQ     76$             ;BR IF YES
022762  104006                                  HLT     6               ;BRANCH POINTS WRONG
022764  104412                          76$:    MSTCLR
022766  012777  000010  156366                  MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
022774  012777  006400  156376                  MOV     #BIT11+BIT10+BIT8,@DVSFR        ;RESET DV11
023002  017704  156362                          MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
023006  042704  177774                          BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
023012  012705  000003                          MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
023016  020504                                  CMP     R5,R4           ;SET EXPECTED
023020  001401                                  BEQ     77$             ;BR POINT OK?
023022  104006                                  HLT     6               ;BR IF YES
023024                                  77$:                            ;BR POINTS WRONG
                                                ;*BRANCH "A" TEST OF ALU 04
                                                ;*
023024  104412                                  MSTCLR                          ;RESET DV11
023026  012777  000010  156326                  MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
023034  012777  000020  156334                  MOV     #BIT4,@DVSRA    ;LOAD DATA
023042  012777  020000  156330                  MOV     #RAM,@DVSFR     ;DO A "RAM READ"
023050  104415                                  ROMCLK                          ;EXECUTE
023052  012777  030261  156320                  MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
023060  104415                                  ROMCLK                          ;XFR RAM OUTPUT TO "A" REG
023062  012777  010037  156310                  MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
023070  104415                                  ROMCLK                          ;F=A
023072  012777  007000  156300                  MOV     #BIT11+BIT10+BIT9,@DVSFR        ;LOAD BRANCH POINT
023100  017704  156264                          MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
023104  042704  177774                          BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
023110  012705  000002                          MOV     #BIT1,R5        ;SET EXPECTED
023114  020504                                  CMP     R5,R4           ;BR POINTS CORRECT?
023116  001401                                  BEQ     78$             ;BR IF YES
023120  104006                                  HLT     6               ;BRANCH POINTS WRONG
023122  104412                          78$:    MSTCLR
023124  012777  000010  156230                  MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
023132  012777  007000  156240                  MOV     #BIT11+BIT10+BIT9,@DVSFR        ;RESET DV11
023140  017704  156224                          MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
023144  042704  177774                          BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
023150  012705  000003                          MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
023154  020504                                  CMP     R5,R4           ;SET EXPECTED
023156  001401                                  BEQ     79$             ;BR POINT OK?
023160  104006                                  HLT     6               ;BR IF YES
023162                                  79$:                            ;BR POINTS WRONG
```

```
 4654  023162  104400                          SCOPE
 4655
 4656
 4657                                          ;************************* TEST 101 *******************************
 4658                                          ;*TEST OF BRANCH 'B'' 'RAM OUTPUT 0-14=0''.
 4659                                          ;*TEST TO A RAM READ AND 'FLOAT'' A '1'' FROM
 4660                                          ;*RAM 0 TO 14 ; EXPECTING 'RAM 014=0'' TO BE FALSE.
 4661                                          ;*THEN THE '1'' IS SHIFTED INTO BIT15 AND
 4662                                          ;*'RAM 0-14=0'' SHOULD BE FALSE.
 4663                                          ;*THIS ALSO TEST 'BRB'' [RAM OUTPUT BIT15] TRUE.
 4664                                          ;:*************************************************************************
 4665
 4666                                  ;  TEST 101
 4667                                  ;--------------
 4668  023164  012737  000101  001226  TST101: MOV     #101,TSTNO
 4669  023172  012737  023364  001216          MOV     #TST102,NEXT
 4670  023200  104412                          MSTCLR                          ;RESET DV11
 4671  023202  012703  000001                  MOV     #1,R3                   ;SET DATA
 4672  023206  012702  076000                  MOV     #BRB+BIT11+BIT10,R2 ;BRB 'RAM OUTPUT 0-14=0''?
 4673  023212  012777  000010  156142          MOV     #BIT3,@DVSCR            ;SET SOURCE SEL.
 4674  023220  010277  156154                  MOV     R2,@DVSFR               ;LOAD BRB TEST
 4675  023224  017704  156140                  MOV     @DVLCR,R4               ;READ TEST POINTS
 4676  023230  012705  000001                  MOV     #BIT0,R5                ;SET EXPECTED
 4677  023234  042704  177774                  BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
 4678  023240  020504                          CMP     R5,R4                   ;TRUE?
 4679  023242  001401                          BEQ     .+4                     ;BR IF YES
 4680  023244  104006                          HLT     6                       ;RAM OUTPUT 0-14 NOT TRUE AFTER INIT
 4681  023246  012705  000003                  MOV     #BIT1+BIT0,R5           ;SET EXPECTED
 4682  023252  010377  156120          1$:     MOV     R3,@DVSRA               ;LOAD DATA
 4683  023256  012777  020000  156114          MOV     #RAM,@DVSFR             ;ISSUE RAM READ INSTR
 4684  023264  104415                          ROMCLK                          ;
 4685  023266  010277  156106                  MOV     R2,@DVSFR               ;BRB TEST
 4686  023272  017704  156072                  MOV     @DVLCR,R4               ;READ BR TEST POINTS
 4687  023276  042704  177774                  BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
 4688  023302  005703                          TST     R3                      ;IS 15=1 IN DATA?
 4689  023304  100002                          BPL     2$                      ;BR IF NO
 4690  023306  042705  000002                  BIC     #BIT1,R5                ;IF 15=1 - 0-14=TRUE
 4691  023312  020504                  2$:     CMP     R5,R4                   ;BRB TEST POINT OK?
 4692  023314  001401                          BEQ     3$                      ;BR IF YES
 4693  023316  104006                          HLT     6                       ;BAD TEST POINT
 4694  023320  000241                  3$:     CLC                             ;CLEAR CARRY
 4695  023322  006103                          ROL     R3                      ;MOV BIT TO NEXT RAM POSITION
 4696  023324  001352                          BNE     1$                      ;HAVE ALL BITS BEEN TESTED?
 4697  023326  012705  000001                  MOV     #BIT0,R5                ;SET EXPECTED RESULTS
 4698  023332  012777  075400  156040          MOV     #BRB+BIT11+BIT9+BIT8,@DVSFR
 4699  023340  017704  156024                  MOV     @DVLCR,R4               ;BRB 'RAM OUTPUT 15H''
 4700  023344  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
 4701  023350  020504                          CMP     R5,R4                   ;BRANCH RESULTS OK?
 4702  023352  001403                          BEQ     4$                      ;BR IF YES
 4703  023354  017702  156020                  MOV     @DVSFR,R2               ;SAVE DVSFR FOR TYPEOUT
 4704  023360  104006                          HLT     6                       ;'RAM OUTPUT 15H'' S/B TRUE
 4705  023362  104400                  4$:     SCOPE                           ;SCOPE TESTS
 4706
 4707
 4708                                          ;************************* TEST 102 *******************************
 4709                                          ;*TEST OF THE RAM WRITE OPERATION.
```

```
                                        ;*WRITE ALL SECONDARY REGISTERS FOR ALL LINES
                                        ;*WITH DIFFERENT DATA BY USING THE ROM
                                        ;*AND VERIFY THE DATA BY THE UNIBUS.
                                        ;;*****************************************************************************

                                ;  TEST 102
                                ;--------------
023364  012737  000102  001226  TST102: MOV     #102,TSTNO
023372  012737  024000  001216          MOV     #TST103,NEXT
023400  104412                        MSTCLR                  ;CLEAR ALL DV11 REGISTERS
023402  012777  000010  155752        MOV     #BIT3,@DVSCR    ;SET SORCE SEC
023410  013700  001400                MOV     DVSFR,R0        ;SET DVSFR POINTER IN R0
023414  012702  010077                MOV     #ALU+BIT5+BIT4+BIT3+BIT2+BIT1+BIT0,R2
                                                                ;FUNCTION 'F=A+1
023420  012703  020760                MOV     #RAM+BIT8+BIT7+BIT6+BIT5+BIT4,R3
                                                                ;RAM WRITE FROM ALU RESULT.
023424  012704  030361                MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,R4
                                                                ;MOVE ALU RESULT TO "A" REG.
023430  005005                        CLR     R5              ;CLEAR LINE NUMBER COUNTER
023432  005001                1$:     CLR     R1              ;ZERO SEC REG POINTER
023434  110577  155732        2$:     MOVB    R5,@DVSRS       ;LOAD LINE
023440  110177  155730                MOVB    R1,@DVSRSH      ;LOAD SEC REG.
023444  012777  177777  155724        MOV     #-1,@DVSRA      ;SET 'FOOT PRINT"
023452  042703  000017                BIC     #17,R3          ;CLEAR LINER
023456  050103                        BIS     R1,R3           ;SET LINE
023460  010310                        MOV     R3,(R0)         ;DO 'RAM WRITE"
023462  104415                        ROMCLK                  ;
023464  012777  077000  155706        MOV     #BRB+BIT11+BIT10+BIT9,@DVSFR
023472  010546                        MOV     R5,-(SP)        ;SAVERS
023474  010446                        MOV     R4,-(SP)        ;SAVE R4
023476  010246                        MOV     R2,-(SP)        ;SAVE R2
023500  012705  000001                MOV     #BIT0,R5        ;EXPECTED
023504  017704  155660                MOV     @DVLCR,R4       ;READ BR. RESULT
023510  042704  177774                BIC     #^C<BIT1+BIT0>,R4 ;STRIP JUNK
023514  020504                        CMP     R5,R4           ;WRITE INHIBIT TRUE?
023516  001401                        BEQ     .+4             ;
023520  104006                        HLT     6               ;WRITE INHIBIT FAILED
023522  022777  177777  155646        CMP     #-1,@DVSRA      ;WAS WRITE
023530  001401                        BEQ     .+4             ;REALLY
023532  104000                        HLT     0               ;INHIBTED?
023534  012602                        MOV     (SP)+,R2        ;RESTORE
023536  012604                        MOV     (SP)+,R4        ;REGISTERS
023540  012605                        MOV     (SP)+,R5        ;
023542  012710  020000                MOV     #RAM,(R0)       ;PLACE RAM INSTR IN SFR
023546  050110                        BIS     R1,(R0)         ;PLACE SEC REG POINTER IN SFR
023550  104415                        ROMCLK                  ;EXECUTE INSTR.
023552  010210                        MOV     R2,(R0)         ;"F=A+1"
023554  104415                        ROMCLK                  ;EXECUTE
023556  042703  000017                BIC     #17,R3          ;CLEAR SEC REG POINTER
023562  050103                        BIS     R1,R3           ;SET SEC REG POINTER
023564  010310                        MOV     R3,(R0)         ;RAM WRITE FROM ALU RESULT
023566  104415                        ROMCLK                  ;EXECUTE
023570  012777  077000  155602        MOV     #BRB+BIT11+BIT10+BIT9,@DVSFR
023576  010546                        MOV     R5,-(SP)        ;TEST WRITE
023600  010446                        MOV     R4,-(SP)        ;INHIBIT
023602  010246                        MOV     R2,-(SP)        ;FALSE!
```

```
4766  023604  017702  155570                  MOV     @DVSFR,R2
4767  023610  012705  000003                  MOV     #BIT1+BIT0,R5   ;EXPECTED
4768  023614  017704  155550                  MOV     @DVLCR,R4       ;READ RESULTS
4769  023620  042704  177774                  BIC     #^C<BIT1+BIT0>.R4 ;STRIP JUNK
4770  023624  020504                          CMP     R5,R4           ;GOOD?
4771  023626  001401                          BEQ     .+4             ;
4772  023630  104006                          HLT     6               ;WRITE INHIBIT S/B FALSE
4773  023632  012602                          MOV     (SP)+,R2        ;RESTORE
4774  023634  012604                          MOV     (SP)+,R4        ;REGISTERS
4775  023636  012605                          MOV     (SP)+,R5        ;
4776  023640  010410                          MOV     R4,(R0)         ;MOVE ALU RESULT TO A REG (UPDATED DATA)
4777  023642  104415                          ROMCLK                  ;EXECUTE
4778  023644  005201                          INC     R1              ;UPDATE SEC REG POINTER
4779  023646  022701  000020                  CMP     #16.,R1         ;ALL REGISTERS DONE?
4780  023652  001270                          BNE     2$              ;BR IF NO
4781  023654  010046                          MOV     R0,-(SP)        ;SAVE R0
4782  023656  004537  031676                  PERFORM ,SETSCAN         ;UPDATE MSCAN
4783  023662  000001                          1                       ;(LINE NO. UPDATE)
4784  023664  012600                          MOV     (SP)+,R0        ;RESTORE R0
4785  023666  012710  030124                  MOV     #XFR+BIT6+BIT4+BIT2,(R0)
4786  023672  104415                          ROMCLK                  ;XFR MSCAN TO RAR 0-3 CLOCK
4787  023674  005205                          INC     R5              ;UPDATE LINE NUMBER COUNTER.
4788  023676  022705  000020                  CMP     #16.,R5         ;ALL LINES DONE?
4789  023702  001253                          BNE     1$              ;BR IF NO
4790  023704  005000                  3$:     CLR     R0              ;CLEAR SEC REG POINTER
4791  023706  005037  001246                  CLR     TEMP1           ;CLEAR LINE NUMBER POINTER
4792  023712  013702  001372                  MOV     DVSRS,R2        ;SET SRC POINTER (LINE SEL)
4793  023716  013703  001374                  MOV     DVSRSH,R3       ;SET HIGH BYTE POINTER (SEC REG SEL)
4794  023722  013701  001376                  MOV     DVSRA,R1        ;SET SRA POINTER (ACCESS REG)
4795  023726  012705  000001                  MOV     #1,R5           ;SET EXPECTED DATA
4796  023732  110013                  4$:     MOVB    R0,(R3)         ;LOAD SEC REG SEL
4797  023734  113712  001246                  MOVB    TEMP1,(R2)      ;LOAD LINE NO.
4798  023740  011104                          MOV     (R1),R4         ;READ RAM RESULT
4799  023742  020504                          CMP     R5,R4           ;WAS RAM ''WRITTEN'' OCCRECTLY?
4800  023744  001401                          BEQ     5$              ;BR IF RAM DATA OK
4801  023746  104006                          HLT     6               ;RAM WAS ''WRITTEN'' INCORRECTLY
4802  023750  122025                  5$:     CMPB    (R0)+,(R5)+     ;UPDATE SEC REG POINTER AND DATA EXPECTED
4803  023752  022700  000020                  CMP     #16.,R0         ;ALL SEC REGISTERS DONE?
4804  023756  001365                          BNE     4$              ;BR IF NO
4805  023760  005000                          CLR     R0              ;ZERO SEC REG POINTER
4806  023762  005237  001246                  INC     TEMP1           ;UPDATE LINE NUMBER POINTER
4807  023766  023727  001246  000020          CMP     TEMP1,#16.      ;ALL LINES DONE?
4808  023774  001356                          BNE     4$              ;BR IF NO
4809  023776  104400                          SCOPE                   ;SCOPE TEST
4810
4811                                          ;************************ TEST 103 *******************************
4812                                          ;*BASIC TEST FOR THE ''BCC OPERATION''
4813                                          ;*POLYNOMIAL SELECTION TABLE:
4814                                          ;*RAM OUTPUT BIT04 BIT03        POLY
4815                                          ;*              0     0         LRC 8
4816                                          ;*              0     1         CRC 16
4817                                          ;*              1     1         CRC CCITT
4818                                          ;:*************************************************************
4819
4820                                          ;************************ TEST 103 *******************************
4821                                          ;*TEST OF LRC 8.
```

```
                                        ;*PATTERNS ARE:
                                        ;*A REG  B REG  BCC (EXPECTED)
                                        ;*  0'S    0'S   0'S
                                        ;*  0'S    1'S   1'S
                                        ;*  1'S    0'S   1'S
                                        ;*  1'S    1'S   0'S
                                        ;;*****************************************************************

                                ;  TEST 103
                                ;-------------
024000  012737  000103  001226  TST103: MOV     #103,TSTNO
024006  012737  024254  001216          MOV     #TST104,NEXT
024014  104412                          MSTCLR                  ;RESET DV11
024016  012777  000010  155336          MOV     #BIT3,@DVSCR    ;SET SORCE SEL
024024  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER
024030  012702  030346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
024034  012710  060000                  MOV     #BCC,(R0)       ;DO A 'BCC' INSTRUCTION USING 'LRC8'
024040  104415                          ROMCLK                  ;EXECUTE
024042  010210                          MOV     R2,(R0)         ;XFR BCC REG TO DVRIC
024044  104415                          ROMCLK                  ;EXECUTE
024046  005005                          CLR     R5              ;SET EXPECTED TO 0
024050  017704  155312                  MOV     @DVRIC,R4       ;READ RESULTS OF BCC REG.
024054  001401                          BEQ     1$              ;BR IF =0
024056  104006                          HLT     6               ;BCC REG NOT =0
024060  012710  010034          1$:     MOV     #ALU+BIT4+BIT3+BIT2,(R0)
024064  104415                          ROMCLK                  ;ALU OPR 'F=-1'
024066  012710  030362                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
024072  104415                          ROMCLK                  ;XFR ALU RESULT TO B REGISTER
024074  012710  060000                  MOV     #BCC,(R0)       ;DO A 'BCC' OPR
024100  104415                          ROMCLK                  ;EXECUTE
024102  010210                          MOV     R2,(R0)         ;XFR BCC REG TO DVRIC
024104  104415                          ROMCLK                  ;EXECUTE
024106  012705  000377                  MOV     #377,R5         ;SET EXPECTED RESULTS =ALL 1'S [LOW BYTE]
024112  017704  155250                  MOV     @DVRIC,R4       ;READ RESULTS
024116  020504                          CMP     R5,R4           ;DID BCC OPR WORK
024120  001401                          BEQ     2$              ;BR IF OK
024122  104006                          HLT     6               ;EITHER 'BCC' OPR FAILED OR BIT(S) DROPPED IN LOW BYTE O
024124  104412                  2$:     MSTCLR                  ;RESET INTERAL REG (BCC,ALU,B)
024126  012777  000010  155226          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
024134  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
024140  104415                          ROMCLK                  ;ALU 'F=-1'
024142  012710  030361                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,(R0)
024146  104415                          ROMCLK                  ;XFR ALU RESULT TO 'A' REGISTER
024150  012710  060000                  MOV     #BCC,(R0)       ;BCC OPERATION
024154  104415                          ROMCLK                  ;EXECUTE
024156  010210                          MOV     R2,(R0)         ;XFR BCC TO DVRIC
024160  104415                          ROMCLK                  ;EXECUTE
024162  017704  155200                  MOV     @DVRIC,R4       ;READ RESULTS
024166  020504                          CMP     R5,R4           ;BCC REG S/B =377
024170  001401                          BEQ     3$              ;BR IF OK
024172  104006                          HLT     6               ;BCC REG NOT =377
                                                                ;READ OF DATA FROM 'A' REG LEG FAILED
024174  104412                  3$:     MSTCLR                  ;CLEAR INTERNAL REGISTERS
024176  012777  000010  155156          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
024204  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
024210  104415                          ROMCLK                  ;ALU 'F=-1'
```

```
4878  024212  012710  030361                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,(R0)
4879  024216  104415                          ROMCLK                  ;XFR ALU RESULT TO "A" REG
4880  024220  012710  030362                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
4881  024224  104415                          ROMCLK                  ;XFR ALU RESULT TO "B" REG
4882  024226  012710  060000                  MOV     #BCC,(R0)       ;SO BCC OPR
4883  024232  104415                          ROMCLK                  ;
4884  024234  010210                          MOV     R2,(R0)         ;XFR BCC TO DVRIC
4885  024236  104415                          ROMCLK                  ;ROMCLK
4886  024240  005005                          CLR     R5              ;EXPECT 0'S
4887  024242  017704  155120                  MOV     @DVRIC,R4       ;READ BCC RESULT
4888  024246  001401                          BEQ     4$              ;BR IF =0
4889  024250  104006                          HLT     6               ;BCC "LRC" XOR TEST FAILED
4890  024252  104400              4$:         SCOPE                   ;SCOPE TEST
4891
4892                                          ;********************** TEST 104 ******************************
4893                                          ;*TEST OF POLYNOMIAL "CRC 16"
4894                                          ;*TEST THAT BITS 9-13 OF THE "B" REG APPEAR
4895                                          ;*IN BITS 1-5 OF THE BCC REG.
4896                                          ;**************************************************************
4897
4898                                  ;  TEST 104
4899                                  ;-------------
4900  024254  012737  000104  001226  TST104: MOV     #104,TSTNO
4901  024262  012737  024446  001216          MOV     #TST105,NEXT
4902  024270  104412                          MSTCLR                  ;RESET DV11
4903  024272  012777  000010  155062          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
4904  024300  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER
4905  024304  012777  001000  155064          MOV     #BIT9,@DVSRA    ;LOAD "DATA"
4906  024312  012710  020000                  MOV     #RAM,(R0)       ;RAM READ
4907  024316  104415                          ROMCLK                  ;EXECUTE
4908  024320  012710  030261                  MOV     #XFR+BIT7+BIT5+BIT4+BIT0,(R0)
4909  024324  104415                          ROMCLK                  ;XFR RAM OUTPUT TO "A" REG.
4910  024326  012710  030262                  MOV     #XFR+BIT7+BIT5+BIT4+BIT1,(R0)
4911  024332  104415                          ROMCLK                  ;XFR RAM OUTPUT TO "B" REG.
4912  024334  012777  000010  155034          MOV     #BIT3,@DVSRA    ;SELECT POLYNOMIAL "CRC16"
4913  024342  012710  020000                  MOV     #RAM,(R0)       ;READ DATA
4914  024346  104415                          ROMCLK                  ;
4915  024350  012705  001000          1$:     MOV     #BIT9,R5        ;SET EXPECTED RESULTS
4916  024354  012710  060000          2$:     MOV     #BCC,(R0)       ;BCC OPR "CRC16"
4917  024360  104415                          ROMCLK                  ;EXECUTE
4918  024362  012710  030346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,(R0)
4919  024366  104415                          ROMCLK                  ;XFR BCC REG TO DVRIC
4920  024370  017704  154772                  MOV     @DVRIC,R4       ;READ RESULTS
4921  024374  012703  000010                  MOV     #8.,R3          ;PREPARE
4922  024400  000241                  3$:     CLC                     ;TO
4923  024402  006104                          ROL     R4              ;POSITION
4924  024404  005303                          DEC     R3              ;RESULT
4925  024406  001374                          BNE     3$              ;OF BCC OPERATION
4926  024410  020504                          CMP     R5,R4           ;DID CRC16 WORK?
4927  024412  001401                          BEQ     4$              ;BR IF YES
4928  024414  104006                          HLT     6               ;INCORRECT BCC RESULTS
4929  024416  012710  010026          4$:     MOV     #ALU+BIT4+BIT2+BIT1,(R0)
4930  024422  104415                          ROMCLK                  ;ALU "F=A+B"
4931  024424  012710  030362                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
4932  024430  104415                          ROMCLK                  ;XFR ALU RESULT TO "B" REG
4933  024432  062705  001000                  ADD     #BIT9,R5        ;UPDATE DATA COMPARE
```

```
4934  024436  032705  040000              BIT     #BIT14,R5       ;ALL DATA DONE?
4935  024442  001744                      BEQ     2$              ;BR IF NO
4936  024444  104400                      SCOPE                   ;SCOPE TEST
4937
4938
4939
4940                                      ;************************ TEST 105 ******************************
4941                                      ;*TEST OF THE BCC OPERATION USING
4942                                      ;*USING CRC16 FOR THE POLYNOMIAL
4943                                      ;*SPECIFIC DATA PATTERNS ARE USED TO
4944                                      ;*ISOLATE FAULTS AS SOON AS POSSIBLE
4945                                      ;:**************************************************************
4946
4947                                  ;  TEST 105
4948                                  ;--------------
4949  024446  012737  000105  001226  TST105: MOV     #105,TSTNO
4950  024454  012737  026102  001216          MOV     #TST106,NEXT
4951  024462  012737  120001  032202          MOV     #CRC16,XPOLY    ;SET POLYNOMIAL
4952  024470  013700  001400                  MOV     DVSFR,R0        ;SET REGISTER POINTER
4953  024474  012702  030346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
4954                                                                  ;SET FOR XFER FROM BCC TO DVRIC
4955  024500  012737  024506  001220          MOV     #1$,LOCK        ;SET IF SW09=1
4956  024506  104412                  1$:     MSTCLR                  ;ISSUE A MSTCLR
4957  024510  012777  000010  154644          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
4958  024516  005037  032206                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
4959  024522  004537  032030                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
4960  024526  000010                          8.                      ;SHIFTS PERFORMED
4961  024530  000000                          0                       ;DATA CHAR
4962  024532  000000                          0                       ;PREVIOUS BCC
4963  024534  004337  031754                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
4964  024540  000000                          0                       ;DATA TO THE A REGISTER
4965  024542  000000                          0                       ;DATA TO THE B REGISTER
4966  024544  000010                          10                      ;POLY SELT. (RAMOUTPUT)
4967  024546  012710  060000                  MOV     #BCC,(R0)
4968  024552  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
4969  024554  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
4970  024556  104415                          ROMCLK                  ;
4971  024560  013705  032206                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
4972  024564  017704  154576                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
4973  024570  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
4974  024572  001401                          BEQ     64$             ;BR IF OK!
4975  024574  104006                          HLT     6               ;BCC CALCULATION ERROR
4976  024576  104401                  64$:    SCOP1                   ;SW09=1?
4977  024600  012737  024606  001220          MOV     #2$,LOCK        ;SET IF SW09=1
4978  024606  104412                  2$:     MSTCLR                  ;ISSUE A MSTCLR
4979  024610  012777  000010  154544          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
4980  024616  005037  032206                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
4981  024622  004537  032030                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
4982  024626  000010                          8.                      ;SHIFTS PERFORMED
4983  024630  000252                          ^B<10101010>                    ;DATA CHAR
4984  024632  040000                          BIT14                   ;PREVIOUS BCC
4985  024634  004337  031754                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
4986  024640  000252                          ^B<10101010>                    ;DATA TO THE A REGISTER
4987  024642  040000                          BIT14                   ;DATA TO THE B REGISTER
4988  024644  000010                          10                      ;POLY SELT. (RAMOUTPUT)
4989  024646  012710  060000                  MOV     #BCC,(R0)
```

```
4990  024652  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
4991  024654  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
4992  024656  104415                          ROMCLK                  ;
4993  024660  013705  032206                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
4994  024664  017704  154476                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
4995  024670  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
4996  024672  001401                          BEQ     65$             ;BR IF OK!
4997  024674  104006                          HLT     6               ;BCC CALCULATION ERROR
4998  024676  104401              65$:        SCOP1                   ;SW09=1?
4999  024700  012737  024706  001220          MOV     #3$,LOCK        ;SET IF SW09=1
5000  024706  104412              3$:         MSTCLR                  ;ISSUE A MSTCLR
5001  024710  012777  000010  154444          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
5002  024716  005037  032206                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
5003  024722  004537  032030                  JSR     R5,S!MBCC       ;GO AND CALCULATE A NEW BCC
5004  024726  000010                          8.                      ;SHIFTS PERFORMED
5005  024730  000125                          ^B<01010101>                    ;DATA CHAR
5006  024732  000000                          0                       ;PREVIOUS BCC
5007  024734  004337  031754                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
5008  024740  000125                          ^B<01010101>                    ;DATA TO THE A REGISTER
5009  024742  000000                          0                       ;DATA TO THE B REGISTER
5010  024744  000010                          10                      ;POLY SELT. (RAMOUTPUT)
5011  024746  012710  060000                  MOV     #BCC,(R0)
5012  024752  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
5013  024754  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
5014  024756  104415                          ROMCLK                  ;
5015  024760  013705  032206                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
5016  024764  017704  154376                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
5017  024770  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
5018  024772  001401                          BEQ     66$             ;BR IF OK!
5019  024774  104006                          HLT     6               ;BCC CALCULATION ERROR
5020  024776  104401              66$:        SCOP1                   ;SW09=1?
5021  025000  012737  025006  001220          MOV     #4$,LOCK        ;SET IF SW09=1
5022  025006  104412              4$:         MSTCLR                  ;ISSUE A MSTCLR
5023  025010  012777  000010  154344          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
5024  025016  005037  032206                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
5025  025022  004537  032030                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
5026  025026  000010                          8.                      ;SHIFTS PERFORMED
5027  025030  000377                          ^B<11111111>                    ;DATA CHAR
5028  025032  000000                          0                       ;PREVIOUS BCC
5029  025034  004337  031754                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
5030  025040  000377                          ^B<11111111>                    ;DATA TO THE A REGISTER
5031  025042  000000                          0                       ;DATA TO THE B REGISTER
5032  025044  000010                          10                      ;POLY SELT. (RAMOUTPUT)
5033  025046  012710  060000                  MOV     #BCC,(R0)
5034  025052  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
5035  025054  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
5036  025056  104415                          ROMCLK                  ;
5037  025060  013705  032206                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
5038  025064  017704  154276                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
5039  025070  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
5040  025072  001401                          BEQ     67$             ;BR IF OK!
5041  025074  104006                          HLT     6               ;BCC CALCULATION ERROR
5042  025076  104401              67$:        SCOP1                   ;SW09=1?
5043  025100  012737  025106  001220          MOV     #5$,LOCK        ;SET IF SW09=1
5044  025106  104412              5$:         MSTCLR                  ;ISSUE A MSTCLR
5045  025110  012777  000010  154244          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
```

```
 5046  025116  005037  032206                    CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
 5047  025122  004537  032030                    JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
 5048  025126  000010                            8.                      ;SHIFTS PERFORMED
 5049  025130  000001                            ^B<00000001>                    ;DATA CHAR
 5050  025132  000000                            0                       ;PREVIOUS BCC
 5051  025134  004337  031754                    JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
 5052  025140  000001                            ^B<00000001>                    ;DATA TO THE A REGISTER
 5053  025142  000000                            0                       ;DATA TO THE B REGISTER
 5054  025144  000010                            10                      ;POLY SELT. (RAMOUTPUT)
 5055  025146  012710  060000                    MOV     #BCC,(R0)
 5056  025152  104415                            ROMCLK                  ;DO A BCC CALCULATION (HRDW)
 5057  025154  010210                            MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
 5058  025156  104415                            ROMCLK                  ;
 5059  025160  013705  032206                    MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
 5060  025164  017704  154176                    MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
 5061  025170  020504                            CMP     R5,R4           ;DOES EXPECTED = FOUND??
 5062  025172  001401                            BEQ     68$             ;BR IF OK!
 5063  025174  104006                            HLT     6               ;BCC CALCULATION ERROR
 5064  025176  104401                    68$:    SCOP1                   ;SW09=1?
 5065  025200  012737  025206  001220            MOV     #6$,LOCK        ;SET IF SW09=1
 5066  025206  104412                    6$:     MSTCLR                  ;ISSUE A MSTCLR
 5067  025210  012777  000010  154144            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 5068  025216  005037  032206                    CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
 5069  025222  004537  032030                    JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
 5070  025226  000010                            8.                      ;SHIFTS PERFORMED
 5071  025230  000004                            ^B<00000100>                    ;DATA CHAR
 5072  025232  000000                            0                       ;PREVIOUS BCC
 5073  025234  004337  031754                    JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
 5074  025240  000004                            ^B<00000100>                    ;DATA TO THE A REGISTER
 5075  025242  000000                            0                       ;DATA TO THE B REGISTER
 5076  025244  000010                            10                      ;POLY SELT. (RAMOUTPUT)
 5077  025246  012710  060000                    MOV     #BCC,(R0)
 5078  025252  104415                            ROMCLK                  ;DO A BCC CALCULATION (HRDW)
 5079  025254  010210                            MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
 5080  025256  104415                            ROMCLK                  ;
 5081  025260  013705  032206                    MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
 5082  025264  017704  154076                    MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
 5083  025270  020504                            CMP     R5,R4           ;DOES EXPECTED = FOUND??
 5084  025272  001401                            BEQ     69$             ;BR IF OK!
 5085  025274  104006                            HLT     6               ;BCC CALCULATION ERROR
 5086  025276  104401                    69$:    SCOP1                   ;SW09=1?
 5087  025300  012737  025306  001220            MOV     #7$,LOCK        ;SET IF SW09=1
 5088  025306  104412                    7$:     MSTCLR                  ;ISSUE A MSTCLR
 5089  025310  012777  000010  154044            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 5090  025316  005037  032206                    CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
 5091  025322  004537  032030                    JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
 5092  025326  000010                            8.                      ;SHIFTS PERFORMED
 5093  025330  000020                            ^B<00010000>                    ;DATA CHAR
 5094  025332  000000                            0                       ;PREVIOUS BCC
 5095  025334  004337  031754                    JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
 5096  025340  000020                            ^B<00010000>                    ;DATA TO THE A REGISTER
 5097  025342  000000                            0                       ;DATA TO THE B REGISTER
 5098  025344  000010                            10                      ;POLY SELT. (RAMOUTPUT)
 5099  025346  012710  060000                    MOV     #BCC,(R0)
 5100  025352  104415                            ROMCLK                  ;DO A BCC CALCULATION (HRDW)
 5101  025354  010210                            MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
```

```
 5102  025356  104415                          ROMCLK                  ;
 5103  025360  013705  032206                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
 5104  025364  017704  153776                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
 5105  025370  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
 5106  025372  001401                          BEQ     70$             ;BR IF OK!
 5107  025374  104006                          HLT     6               ;BCC CALCULATION ERROR
 5108  025376  104401              70$:        SCOP1                   ;SW09=1?
 5109  025400  012737  025406  001220          MOV     #8$,LOCK        ;SET IF SW09=1
 5110  025406  104412              8$:         MSTCLR                  ;ISSUE A MSTCLR
 5111  025410  012777  000010  153744          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 5112  025416  005037  032206                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
 5113  025422  004537  032030                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
 5114  025426  000010                          8.                      ;SHIFTS PERFORMED
 5115  025430  000100                          ^B<01000000>                    ;DATA CHAR
 5116  025432  000000                          0                       ;PREVIOUS BCC
 5117  025434  004337  031754                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
 5118  025440  000100                          ^B<01000000>                    ;DATA TO THE A REGISTER
 5119  025442  000000                          0                       ;DATA TO THE B REGISTER
 5120  025444  000010                          10                      ;POLY SELT. (RAMOUTPUT)
 5121  025446  012710  060000                  MOV     #BCC,(R0)
 5122  025452  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
 5123  025454  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
 5124  025456  104415                          ROMCLK                  ;
 5125  025460  013705  032206                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
 5126  025464  017704  153676                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
 5127  025470  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
 5128  025472  001401                          BEQ     71$             ;BR IF OK!
 5129  025474  104006                          HLT     6               ;BCC CALCULATION ERROR
 5130  025476  104401              71$:        SCOP1                   ;SW09=1?
 5131  025500  012737  025506  001220          MOV     #9$,LOCK        ;SET IF SW09=1
 5132  025506  104412              9$:         MSTCLR                  ;ISSUE A MSTCLR
 5133  025510  012777  000010  153644          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 5134  025516  005037  032206                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
 5135  025522  004537  032030                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
 5136  025526  000010                          8.                      ;SHIFTS PERFORMED
 5137  025530  000006                          ^B<00000110>                    ;DATA CHAR
 5138  025532  000000                          0                       ;PREVIOUS BCC
 5139  025534  004337  031754                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
 5140  025540  000006                          ^B<00000110>                    ;DATA TO THE A REGISTER
 5141  025542  000000                          0                       ;DATA TO THE B REGISTER
 5142  025544  000010                          10                      ;POLY SELT. (RAMOUTPUT)
 5143  025546  012710  060000                  MOV     #BCC,(R0)
 5144  025552  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
 5145  025554  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
 5146  025556  104415                          ROMCLK                  ;
 5147  025560  013705  032206                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
 5148  025564  017704  153576                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
 5149  025570  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
 5150  025572  001401                          BEQ     72$             ;BR IF OK!
 5151  025574  104006                          HLT     6               ;BCC CALCULATION ERROR
 5152  025576  104401              72$:        SCOP1                   ;SW09=1?
 5153  025600  012737  025606  001220          MOV     #100$,LOCK      ;SET IF SW09=1
 5154  025606  104412              100$:       MSTCLR                  ;ISSUE A MSTCLR
 5155  025610  012777  000010  153544          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 5156  025616  005037  032206                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
 5157  025622  004537  032030                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
```

```
5158  025626  000010                                8.                      ;SHIFTS PERFORMED
5159  025630  000120                                ^B<01010000>                    ;DATA CHAR
5160  025632  000000                                0                       ;PREVIOUS BCC
5161  025634  004337  031754                        JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
5162  025640  000120                                ^B<01010000>                    ;DATA TO THE A REGISTER
5163  025642  000000                                0                       ;DATA TO THE B REGISTER
5164  025644  000010                                10                      ;POLY SELT. (RAMOUTPUT)
5165  025646  012710  060000                        MOV     #BCC,(R0)
5166  025652  104415                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
5167  025654  010210                                MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
5168  025656  104415                                ROMCLK                  ;
5169  025660  013705  032206                        MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
5170  025664  017704  153476                        MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
5171  025670  020504                                CMP     R5,R4           ;DOES EXPECTED = FOUND??
5172  025672  001401                                BEQ     73$             ;BR IF OK!
5173  025674  104006                                HLT     6               ;BCC CALCULATION ERROR
5174  025676  104401                        73$:    SCOP1                   ;SW09=1?
5175  025700  012737  025706  001220                MOV     #101$,LOCK      ;SET IF SW09=1
5176  025706  104412                        101$:   MSTCLR                  ;ISSUE A MSTCLR
5177  025710  012777  000010  153444                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
5178  025716  005037  032206                        CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
5179  025722  004537  032030                        JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
5180  025726  000010                                8.                      ;SHIFTS PERFORMED
5181  025730  000320                                ^B<11010000>                    ;DATA CHAR
5182  025732  000000                                0                       ;PREVIOUS BCC
5183  025734  004337  031754                        JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
5184  025740  000320                                ^B<11010000>                    ;DATA TO THE A REGISTER
5185  025742  000000                                0                       ;DATA TO THE B REGISTER
5186  025744  000010                                10                      ;POLY SELT. (RAMOUTPUT)
5187  025746  012710  060000                        MOV     #BCC,(R0)
5188  025752  104415                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
5189  025754  010210                                MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
5190  025756  104415                                ROMCLK                  ;
5191  025760  013705  032206                        MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
5192  025764  017704  153376                        MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
5193  025770  020504                                CMP     R5,R4           ;DOES EXPECTED = FOUND??
5194  025772  001401                                BEQ     74$             ;BR IF OK!
5195  025774  104006                                HLT     6               ;BCC CALCULATION ERROR
5196  025776  104401                        74$:    SCOP1                   ;SW09=1?
5197  026000  012737  026006  001220                MOV     #102$,LOCK      ;SET IF SW09=1
5198  026006  104412                        102$:   MSTCLR                  ;ISSUE A MSTCLR
5199  026010  012777  000010  153344                MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
5200  026016  005037  032206                        CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
5201  026022  004537  032030                        JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
5202  026026  000010                                8.                      ;SHIFTS PERFORMED
5203  026030  000300                                ^B<11000000>                    ;DATA CHAR
5204  026032  000000                                0                       ;PREVIOUS BCC
5205  026034  004337  031754                        JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
5206  026040  000300                                ^B<11000000>                    ;DATA TO THE A REGISTER
5207  026042  000000                                0                       ;DATA TO THE B REGISTER
5208  026044  000010                                10                      ;POLY SELT. (RAMOUTPUT)
5209  026046  012710  060000                        MOV     #BCC,(R0)
5210  026052  104415                                ROMCLK                  ;DO A BCC CALCULATION (HRDW)
5211  026054  010210                                MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
5212  026056  104415                                ROMCLK                  ;
5213  026060  013705  032206                        MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
```

```
5214  026064  017704  153276                MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
5215  026070  020504                        CMP     R5,R4           ;DOES EXPECTED = FOUND??
5216  026072  001401                        BEQ     75$             ;BR IF OK!
5217  026074  104006                        HLT     6               ;BCC CALCULATION ERROR
5218  026076  104401                75$:    SCOP1                   ;SW09=1?
5219  026100  104400                        SCOPE
5220
5221
5222                                        ;************************ TEST 106 *******************************
5223                                        ;*TEST TO RUN A BINARY COUNT (000-377)PATTERN
5224                                        ;*THROUGH THE BCC GENERATION LOGIC.
5225                                        ;*THE POLYNOMIAL USED WILL BE LRC8 .
5226                                        ;*THE BCC REGISTER WILL BE BUILT UP AFTER
5227                                        ;*EACH CHARACTER -*NOT ZEROED OUT*-
5228                                        ;:*************************************************************************
5229
5230                                ;  TEST 106
5231                                ;-------------
5232  026102  012737  000106  001226 TST106: MOV     #106,TSTNO
5233  026110  012737  026260  001216         MOV     #TST107,NEXT
5234  026116  012737  000200  032202         MOV     #LRC8,XPOLY     ;LOAD POLYNOMIAL FOR SOFTWARE CAL.
5235  026124  012702  030346                MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
5236  026130  013700  001400                MOV     DVSFR,R0        ;LOAD DATA XFER FROM BCC TO DVRIC
5237  026134  005001                        CLR     R1              ;SET DATA POINTER TO 0
5238  026136  005037  032206                CLR     CALBCC          ;ZERO SOFTWARE BCC
5239  026142  104412                1$:     MSTCLR                  ;CLEAR THE DV11
5240  026144  012777  000010  153210        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
5241  026152  010137  026204                MOV     R1,2$           ;LOAD SOFTWARE CHAR
5242  026156  010137  026214                MOV     R1,4$           ;LOAD HRDW CHAR
5243  026162  013737  032206  026206        MOV     CALBCC,3$       ;PLACE PREVIOUS BCC FOR SOFTWARE
5244  026170  013737  032206  026216        MOV     CALBCC,5$       ;PLACE PREVIOUS BCC FOR HRDW
5245  026176  004537  032030                JSR     R5,SIMBCC       ;HAVE SOFTWARE GET THE RIGHT BCC
5246  026202  000010                        8.                      ;EIGHT SHIFTS
5247  026204  000001                2$:     .BLKW 1                 ;DATA
5248  026206  000001                3$:     .BLKW 1                 ;PREVIOS BCC
5249  026210  004337  031754                JSR     R3,L.DATA       ;LOAD DV11 REGISTERS
5250  026214  000001                4$:     .BLKW 1                 ;TO BE PLACED INTO THE "A" REG
5251  026216  000001                5$:     .BLKW 1                 ;TO BE PLACED INTO THE "B" REG
5252  026220  000000                        0                       ;TO BE LEFT IN THE RAM OUTPUT REG
5253  026222  012710  060000                MOV     #BCC,(R0)
5254  026226  104415                        ROMCLK                  ;DO A BCC OPERATION
5255  026230  010210                        MOV     R2,(R0)         ;DO A DATA XFER OPR.
5256  026232  104415                        ROMCLK                  ;
5257  026234  013705  032206                MOV     CALBCC,R5       ;GET GOOD BCC
5258  026240  017704  153122                MOV     @DVRIC,R4       ;GET ???? BCC
5259  026244  020504                        CMP     R5,R4           ;ARE THEY THE SAME?
5260  026246  001401                        BEQ     6$              ;BR IF YES
5261  026250  104006                        HLT     6               ;BCC ERROR
5262  026252  105201                6$:     INCB    R1              ;UPDATE DATA CHAR
5263  026254  001332                        BNE     1$              ;BR IF NOT ALL DATA DONE.
5264  026256  104400                        SCOPE                   ;SCOPE THIS TEST
5265
5266
5267                                        ;************************ TEST 107 *******************************
5268                                        ;*TEST TO RUN A BINARY COUNT (000-377)PATTERN
5269                                        ;*THROUGH THE BCC GENERATION LOGIC.
```

```
                                        ;*THE POLYNOMIAL USED WILL BE CRC16 .
                                        ;*THE BCC REGISTER WILL BE BUILT UP AFTER
                                        ;*EACH CHARACTER -*NOT ZEROED OUT*-
                                        ;;*****************************************************************

                                ;  TEST 107
                                ;-------------
026260  012737  000107  001226  TST107: MOV     #107,TSTNO
026266  012737  026436  001216          MOV     #TST110,NEXT
026274  012737  120001  032202          MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL FOR SOFTWARE CAL.
026302  012702  030346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
026306  013700  001400                  MOV     DVSFR,R0        ;LOAD DATA XFER FROM BCC TO DVRIC
026312  005001                          CLR     R1              ;SET DATA POINTER TO 0
026314  005037  032206                  CLR     CALBCC          ;ZERO SOFTWARE BCC
026320  104412                  1$:     MSTCLR                  ;CLEAR THE DV11
026322  012777  000010  153032          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
026330  010137  026362                  MOV     R1,2$           ;LOAD SOFTWARE CHAR
026334  010137  026372                  MOV     R1,4$           ;LOAD HRDW CHAR
026340  013737  032206  026364          MOV     CALBCC,3$       ;PLACE PREVIOUS BCC FOR SOFTWARE
026346  013737  032206  026374          MOV     CALBCC,5$       ;PLACE PREVIOUS BCC FOR HRDW
026354  004537  032030                  JSR     R5,SIMBCC       ;HAVE SOFTWARE GET THE RIGHT BCC
026360  000010                          8.                      ;EIGHT SHIFTS
026362  000001                  2$:     .BLKW 1                 ;DATA
026364  000001                  3$:     .BLKW 1                 ;PREVIOS BCC
026366  004337  031754                  JSR     R3,L.DATA       ;LOAD DV11 REGISTERS
026372  000001                  4$:     .BLKW 1                 ;TO BE PLACED INTO THE "A" REG
026374  000001                  5$:     .BLKW 1                 ;TO BE PLACED INTO THE "B" REG
026376  000010                          10                      ;TO BE LEFT IN THE RAM OUTPUT REG
026400  012710  060000                  MOV     #BCC,(R0)
026404  104415                          ROMCLK                  ;DO A BCC OPERATION
026406  010210                          MOV     R2,(R0)         ;DO A DATA XFER OPR.
026410  104415                          ROMCLK                  ;
026412  013705  032206                  MOV     CALBCC,R5       ;GET GOOD BCC
026416  017704  152744                  MOV     @DVRIC,R4       ;GET ???? BCC
026422  020504                          CMP     R5,R4           ;ARE THEY THE SAME?
026424  001401                          BEQ     6$              ;BR IF YES
026426  104006                          HLT     6               ;BCC ERROR
026430  105201                  6$:     INCB    R1              ;UPDATE DATA CHAR
026432  001332                          BNE     1$              ;BR IF NOT ALL DATA DONE.
026434  104400                          SCOPE                   ;SCOPE THIS TEST

                                        ;************************* TEST 110 ******************************
                                        ;*TEST TO RUN A BINARY COUNT (000-377)PATTERN
                                        ;*THROUGH THE BCC GENERATION LOGIC.
                                        ;*THE POLYNOMIAL USED WILL BE CRC.CCITT .
                                        ;*THE BCC REGISTER WILL BE BUILT UP AFTER
                                        ;*EACH CHARACTER -*NOT ZEROED OUT*-
                                        ;;*****************************************************************

                                ;  TEST 110
                                ;-------------
026436  012737  000110  001226  TST110: MOV     #110,TSTNO
026444  012737  026614  001216          MOV     #TST111,NEXT
026452  012737  102010  032202          MOV     #CRC.CCITT,XPOLY        ;LOAD POLYNOMIAL FOR SOFTWARE CAL.
026460  012702  030346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
```

```
5326  026464  013700  001400                    MOV     DVSFR,R0        ;LOAD DATA XFER FROM BCC TO DVRIC
5327  026470  005001                            CLR     R1              ;SET DATA POINTER TO 0
5328  026472  005037  032206                    CLR     CALBCC          ;ZERO SOFTWARE BCC
5329  026476  104412                    1$:     MSTCLR                  ;CLEAR THE DV11
5330  026500  012777  000010  152654            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
5331  026506  010137  026540                    MOV     R1,2$           ;LOAD SOFTWARE CHAR
5332  026512  010137  026550                    MOV     R1,4$           ;LOAD HRDW CHAR
5333  026516  013737  032206  026542            MOV     CALBCC,3$       ;PLACE PREVIOUS BCC FOR SOFTWARE
5334  026524  013737  032206  026552            MOV     CALBCC,5$       ;PLACE PREVIOUS BCC FOR HRDW
5335  026532  004537  032030                    JSR     R5,SIMBCC       ;HAVE SOFTWARE GET THE RIGHT BCC
5336  026536  000010                            8.                      ;EIGHT SHIFTS
5337  026540  000001                    2$:     .BLKW 1                 ;DATA
5338  026542  000001                    3$:     .BLKW 1                 ;PREVIOS BCC
5339  026544  004337  031754                    JSR     R3,L.DATA       ;LOAD DV11 REGISTERS
5340  026550  000001                    4$:     .BLKW 1                 ;TO BE PLACED INTO THE "A" REG
5341  026552  000001                    5$:     .BLKW 1                 ;TO BE PLACED INTO THE "B" REG
5342  026554  000030                            30                      ;TO BE LEFT IN THE RAM OUTPUT REG
5343  026556  012710  060000                    MOV     #BCC,(R0)
5344  026562  104415                            ROMCLK                  ;DO A BCC OPERATION
5345  026564  010210                            MOV     R2,(R0)         ;DO A DATA XFER OPR.
5346  026566  104415                            ROMCLK                  ;
5347  026570  013705  032206                    MOV     CALBCC,R5       ;GET GOOD BCC
5348  026574  017704  152566                    MOV     @DVRIC,R4       ;GET ???? BCC
5349  026600  020504                            CMP     R5,R4           ;ARE THEY THE SAME?
5350  026602  001401                            BEQ     6$              ;BR IF YES
5351  026604  104006                            HLT     6               ;BCC ERROR
5352  026606  105201                    6$:     INCB    R1              ;UPDATE DATA CHAR
5353  026610  001332                            BNE     1$              ;BR IF NOT ALL DATA DONE.
5354  026612  104400                            SCOPE                   ;SCOPE THIS TEST
5355
```

```
                                        ;************************ TEST 111 *******************************
                                        ;*TEST THAT SETTING BIT9!BIT7 AND BIT9!BIT6
                                        ;*RECV IE AND RECV INTR PRODUCE AN INTERUPT ON VECTOR 'A'
                                        ;:*************************************************************

                                ;  TEST 111
                                ;--------------
026614  012737  000111  001226 TST111: MOV     #111,TSTNO
026622  012737  026764  001216         MOV     #TST112,NEXT
026630  012737  000340  177776         MOV     #340,PS          ;LOCK OUT CPU INTERUPTS
026636  104412                         MSTCLR                   ;ISSUE DVRESET
026640  004537  032210                 JSR     R5,SETVEC        ;GO SET VECTOR 'A' AND 'B'
026644  032232                         NO.ATRAP                 ;'A'
026646  032236                         NO.BTRAP                 ;'B'
026650    340     340                  .BYTE   340,340          ;PRIO. AT 7
026652  005037  177776                 CLR     PS               ;ZERO CPU PRIO.
026656  012777  001200  152476         MOV     #BIT9!BIT7,@DVSCR        ;SET AN INTERUPT RELATIVE BIT.
026664  000240                         NOP                      ;WAIST TIME
026666  012777  001100  152466         MOV     #BIT9!BIT6,@DVSCR        ;SET THE ALTERNATE RELATIVE BIT.
026674  000240                         NOP                      ;WAIST
026676  005077  152460                 CLR     @DVSCR           ;ZERO REG
026702  004537  032210                 JSR     R5,SETVEC        ;GO RESET VECTORS 'A' AND 'B'
026706  026746                         2$                       ;'A'
026710  026760                         3$                       ;'B'
026712    340     340                  .BYTE   340,340          ;PRIO. AT 7
026714  052777  001300  152440         BIS     #BIT9!BIT7!BIT9!BIT6,@DVSCR
026722  000240                         NOP                      ;SET BOTH INTERUPT RELATIVE BITS.
026724  104007                         HLT     7                ;SETTING OF THESE BITS FAILED TO PRODUCE AN INTERUPT
026726  004537  032210          1$:    JSR     R5,SETVEC        ;RESET VECTORS
026732  032232                         NO.ATRAP                 ;'A'
026734  032236                         NO.BTRAP                 ;'B'
026736    340     340                  .BYTE   340,340
026740  005077  152416                 CLR     @DVSCR           ;DSABLE DV11
026744  104400                         SCOPE                    ;SCOPE THIS TEST.
026746  012706  001200          2$:    MOV     #STACK,SP        ;RESET STACK
026752  005077  152404                 CLR     @DVSCR           ;DSABLE DV11
026756  000763                         BR      1$               ;RETURN
026760  104012          3$:            HLT     12               ;VECTOR HERE WAS WRONG SIDE
026762  000771                         BR      2$

                                        ;************************ TEST 112 *******************************
                                        ;*TEST THAT SETTING BIT12 AND BIT10
                                        ;*STORE IE AND NPR STAT OVFLOW PRODUCE AN INTERUPT ON VECTOR 'B'
                                        ;:*************************************************************

                                ;  TEST 112
                                ;--------------
026764  012737  000112  001226 TST112: MOV     #112,TSTNO
026772  012737  027134  001216         MOV     #TST113,NEXT
027000  012737  000340  177776         MOV     #340,PS          ;LOCK OUT CPU INTERUPTS
027006  104412                         MSTCLR                   ;ISSUE DVRESET
027010  004537  032210                 JSR     R5,SETVEC        ;GO SET VECTOR 'A' AND 'B'
027014  032232                         NO.ATRAP                 ;'A'
027016  032236                         NO.BTRAP                 ;'B'
027020    340     340                  .BYTE   340,340          ;PRIO. AT 7
```

```
5412  027022  005037  177776                CLR     PS                      ;ZERO CPU PRIO.
5413  027026  012777  010000  152326        MOV     #BIT12,@DVSCR           ;SET AN INTERUPT RELATIVE BIT.
5414  027034  000240                        NOP                             ;WAIST TIME
5415  027036  012777  002000  152316        MOV     #BIT10,@DVSCR           ;SET THE ALTERNATE RELATIVE BIT.
5416  027044  000240                        NOP                             ;WAIST
5417  027046  005077  152310                CLR     @DVSCR                  ;ZERO REG
5418  027052  004537  032210                JSR     R5,SETVEC               ;GO RESET VECTORS ''A'' AND 'B''
5419  027056  027130                        3$                              ;''A''
5420  027060  027116                        2$                              ;'B''
5421  027062     340     340                .BYTE   340,340                 ;PRIO. AT 7
5422  027064  052777  012000  152270        BIS     #BIT12!BIT10,@DVSCR
5423  027072  000240                        NOP                             ;SET BOTH INTERUPT RELATIVE BITS.
5424  027074  104010                        HLT     10                      ;SETTING OF THESE BITS FAILED TO PRODUCE AN INTERUPT
5425  027076  004537  032210          1$:   JSR     R5,SETVEC               ;RESET VECTORS
5426  027102  032232                        NO.ATRAP                        ;''A''
5427  027104  032236                        NO.BTRAP                        ;'B''
5428  027106     340     340                .BYTE   340,340
5429  027110  005077  152246                CLR     @DVSCR                  ;DSABLE DV11
5430  027114  104400                        SCOPE                           ;SCOPE THIS TEST.
5431  027116  012706  001200          2$:   MOV     #STACK,SP               ;RESET STACK
5432  027122  005077  152234                CLR     @DVSCR                  ;DSABLE DV11
5433  027126  000763                        BR      1$                      ;RETURN
5434  027130  104011                3$:     HLT     11                      ;VECTOR HERE WAS WRONG SIDE
5435  027132  000771                        BR      2$
5436
5437                                        ;************************** TEST 113 ********************************
5438                                        ;*TEST THAT SETTING BIT15!BIT9 AND BIT13!BIT9
5439                                        ;*NPR STAT INTR AND NPR STAT IE PRODUCE AN INTERUPT ON VECTOR 'B''
5440                                        ;;*******************************************************************
5441
5442                                ;  TEST 113
5443                                ;--------------
5444  027134  012737  000113  001226  TST113: MOV     #113,TSTNO
5445  027142  012737  027312  001216        MOV     #TST114,NEXT
5446  027150  012737  000340  177776        MOV     #340,PS                 ;LOCK OUT CPU INTERUPTS
5447  027156  104412                        MSTCLR                          ;ISSUE DVRESET
5448  027160  004537  032210                JSR     R5,SETVEC               ;GO SET VECTOR ''A'' AND 'B''
5449  027164  032232                        NO.ATRAP                        ;''A''
5450  027166  032236                        NO.BTRAP                        ;'B''
5451  027170     340     340                .BYTE   340,340                 ;PRIO. AT 7
5452  027172  005037  177776                CLR     PS                      ;ZERO CPU PRIO.
5453  027176  012777  101000  152156        MOV     #BIT15!BIT9,@DVSCR      ;SET AN INTERUPT RELATIVE BIT.
5454  027204  000240                        NOP                             ;WAIST TIME
5455  027206  012777  021000  152146        MOV     #BIT13!BIT9,@DVSCR      ;SET THE ALTERNATE RELATIVE BIT.
5456  027214  000240                        NOP                             ;WAIST
5457  027216  005077  152140                CLR     @DVSCR                  ;ZERO REG
5458  027222  012777  001000  152132        MOV     #BIT9,@DVSCR            ;SET SYS MAINT ENABLE
5459  027230  004537  032210                JSR     R5,SETVEC               ;GO RESET VECTORS ''A'' AND 'B''
5460  027234  027306                        3$                              ;''A''
5461  027236  027274                        2$                              ;'B''
5462  027240     340     340                .BYTE   340,340                 ;PRIO. AT 7
5463  027242  052777  121000  152112        BIS     #BIT15!BIT9!BIT13!BIT9,@DVSCR
5464  027250  000240                        NOP                             ;SET BOTH INTERUPT RELATIVE BITS.
5465  027252  104010                        HLT     10                      ;SETTING OF THESE BITS FAILED TO PRODUCE AN INTERUPT
5466  027254  004537  032210          1$:   JSR     R5,SETVEC               ;RESET VECTORS
5467  027260  032232                        NO.ATRAP                        ;''A''
```

```
5468  027262  032236                          NO.BTRAP                ;'B'
5469  027264     340     340                  .BYTE   340,340
5470  027266  005077  152070                  CLR     @DVSCR          ;DSABLE DV11
5471  027272  104400                          SCOPE                   ;SCOPE THIS TEST.
5472  027274  012706  001200          2$:     MOV     #STACK,SP       ;RESET STACK
5473  027300  005077  152056                  CLR     @DVSCR          ;DSABLE DV11
5474  027304  000763                          BR      1$              ;RETURN
5475  027306  104011                  3$:     HLT     11              ;VECTOR HERE WAS WRONG SIDE
5476  027310  000771                          BR      2$
5477
5478                                          ;************************ TEST 114 ******************************
5479                                          ;*TEST TO VERIFY THAT VECTOR "A"
5480                                          ;*OCCURES BEFOR VECTOR "B" EVEN
5481                                          ;*WHEN VECTOR "B" IS ENABLED BEFORE
5482                                          ;*VECTOR "A".
5483                                          ;:***************************************************************************
5484
5485                                  ;  TEST 114
5486                                  ;-------------
5487  027312  012737  000114  001226  TST114: MOV     #114,TSTNO
5488  027320  012737  027452  001216          MOV     #TST115,NEXT
5489  027326  012737  000340  177776          MOV     #340,PS         ;SET CPU TO PRIO 7
5490  027334  104412                          MSTCLR                  ;RESET DV11
5491  027336  005003                          CLR     R3              ;SET COUNTER TO 0
5492  027340  012700  000002                  MOV     #2,R0           ;SET TO GET 2 INTERRUPTS
5493  027344  004537  032210                  JSR     R5,SETVEC       ;SET DV11 VECTORS
5494  027350  027420                          2$                      ;RECEIVER INTERRUPTS TO 2$
5495  027352  027442                          3$                      ;TRANSMITTER INTERRUPTS TO 3$
5496  027354     340     340                  .BYTE   340,340         ;SET PRIORITY TO 7 ON INTERRUPT
5497  027356  012777  001000  151776          MOV     #BIT9,@DVSCR    ;SET SYSTEM MAINT
5498  027364  052777  120000  151770          BIS     #BIT15+BIT13,@DVSCR
5499  027372  052777  001300  151762          BIS     #BIT9+BIT7+BIT6,@DVSCR
5500  027400  005037  177776                  CLR     PS              ;SET B INTERRUPT ;A INTERRUPT ;CLEAR CPU STATUS
5501  027404  105203                          INCB    R3              ;HANG WAITING FOR INTERRUPTS
5502  027406  001376                          BNE     .-2             ;
5503  027410  005700                          TST     R0              ;DID TWO INTERRUPTS OCCUR?
5504  027412  001401                          BEQ     1$              ;BR IF YES
5505  027414  104000                          HLT     0               ;EITHER NOT ENOUGH(2) OR TOO MANY (72) INTERRUPTS
5506  027416  104400                  1$:     SCOPE                   ;SCOPE TEST
5507  027420  005300                  2$:     DEC     R0              ;RXISR
5508  027422  022700  000001                  CMP     #1,R0           ;IF RX GOT HERE 1ST R0-1 S/B=1
5509  027426  001401                          BEQ     64$             ;BR IF OK
5510  027430  104011                          HLT     11              ;RX NOT 1ST
5511  027432  042777  000300  151722  64$:    BIC     #BIT7+BIT6,@DVSCR ;DISQUALIFY INTERRUPT REQST
5512  027440  000002                          RTI                     ;RETURN
5513  027442  005300                  3$:     DEC     R0              ;TX ISR
5514  027444  001401                          BEQ     4$              ;IF SCOND INTERRUPT R0 SHOULD NOW =0
5515  027446  104012                          HLT     12              ;TX INTERRUPT OUT OF ORDER
5516  027450  000002                  4$:     RTI                     ;RETURN
5517
5518
5519                                          ;************************ TEST 115 ******************************
5520                                          ;*PRIORITY INTERUPT TESTS.
5521                                          ;*SET PS TO PRIORITY 7 AND VERIFY
5522                                          ;*THAT THE DV11 DOESN'T INTERUPT.
5523                                          ;:***************************************************************************
```

```
                                        ;  TEST 115
                                        ;--------------
027452  012737  000115  001226  TST115: MOV     #115,TSTNO
027460  012737  027542  001216          MOV     #TST116,NEXT
027466  012737  000340  177776          MOV     #340,PS         ;LOCK OUT INTERUPTS
027474  104412                          MSTCLR                  ;CLEAN DV11
027476  004537  032210                  JSR     R5,SETVEC       ;PREPARE VECTORS
027502  032232                          NO.ATRAP                        ;'A'
027504  032236                          NO.BTRAP                ;'B'
027506     340     340                  .BYTE   340,340         ;PRIO. AT 7
027510  012777  001300  151644          MOV     #BIT9+BIT7+BIT6,@DVSCR
027516  012737  000340  177776          MOV     #340,PS ;SET CPU PRIO AND ENABLE VECT. 'A'
027524  000240                          NOP                     ;WAIST
027526  005077  151630          1$:     CLR     @DVSCR          ;DSABLE DV11
027532  104400                          SCOPE                   ;SCOPE THIS TEST
027534  012716  027526          2$:     MOV     #1$,(SP)        ;SET FOR RETURN
027540  000002                          RTI                     ;

                                        ;************************ TEST 116 *******************************
                                        ;*PRIORITY INTERUPT TESTS.
                                        ;*SET PS TO PRIORITY 6 AND VERIFY
                                        ;*THAT THE DV11 DOESN'T INTERUPT.
                                        ;:*****************************************************************

                                        ;  TEST 116
                                        ;--------------
027542  012737  000116  001226  TST116: MOV     #116,TSTNO
027550  012737  027632  001216          MOV     #TST117,NEXT
027556  012737  000340  177776          MOV     #340,PS         ;LOCK OUT INTERUPTS
027564  104412                          MSTCLR                  ;CLEAN DV11
027566  004537  032210                  JSR     R5,SETVEC       ;PREPARE VECTORS
027572  032232                          NO.ATRAP                        ;'A'
027574  032236                          NO.BTRAP                ;'B'
027576     340     340                  .BYTE   340,340         ;PRIO. AT 7
027600  012777  001300  151554          MOV     #BIT9+BIT7+BIT6,@DVSCR
027606  012737  000300  177776          MOV     #300,PS ;SET CPU PRIO AND ENABLE VECT. 'A'
027614  000240                          NOP                     ;WAIST
027616  005077  151540          1$:     CLR     @DVSCR          ;DSABLE DV11
027622  104400                          SCOPE                   ;SCOPE THIS TEST
027624  012716  027616          2$:     MOV     #1$,(SP)        ;SET FOR RETURN
027630  000002                          RTI                     ;

                                        ;************************ TEST 117 *******************************
                                        ;*PRIORITY INTERUPT TESTS.
                                        ;*SET PS TO PRIORITY 5 AND VERIFY
                                        ;*THAT THE DV11 DOESN'T INTERUPT.
                                        ;:*****************************************************************

                                        ;  TEST 117
                                        ;--------------
027632  012737  000117  001226  TST117: MOV     #117,TSTNO
027640  012737  027722  001216          MOV     #TST120,NEXT
027646  012737  000340  177776          MOV     #340,PS         ;LOCK OUT INTERUPTS
```

```
5580  027654  104412                          MSTCLR                          ;CLEAN DV11
5581  027656  004537  032210                  JSR     R5,SETVEC               ;PREPARE VECTORS
5582  027662  032232                          NO.ATRAP                                ;''A''
5583  027664  032236                          NO.BTRAP                        ;''B''
5584  027666     340     340                  .BYTE   340,340                 ;PRIO. AT 7
5585  027670  012777  001300  151464          MOV     #BIT9+BIT7+BIT6,@DVSCR
5586  027676  012737  000240  177776          MOV     #240,PS ;SET CPU PRIO AND ENABLE VECT. ''A''
5587  027704  000240                          NOP                             ;WAIST
5588  027706  005077  151450          1$:     CLR     @DVSCR                  ;DSABLE DV11
5589  027712  104400                          SCOPE                           ;SCOPE THIS TEST
5590  027714  012716  027706          2$:     MOV     #1$,(SP)                ;SET FOR RETURN
5591  027720  000002                          RTI                             ;
5592
5593
5594                                          ;************************ TEST 120 ******************************
5595                                          ;*PRIORITY INTERUPT TESTS.
5596                                          ;*SET PS TO PRIORITY 4 AND VERIFY
5597                                          ;*THAT THE DV11 DOES INTERUPT.
5598                                          ;:*************************************************************************
5599
5600                                  ;  TEST 120
5601                                  ;--------------
5602  027722  012737  000120  001226  TST120: MOV     #120,TSTNO
5603  027730  012737  030014  001216          MOV     #TST121,NEXT
5604  027736  012737  000340  177776          MOV     #340,PS                 ;LOCK OUT INTERUPTS
5605  027744  104412                          MSTCLR                          ;CLEAN DV11
5606  027746  004537  032210                  JSR     R5,SETVEC               ;PREPARE VECTORS
5607  027752  030006                          2$                              ;''A''
5608  027754  032236                          NO.BTRAP                        ;''B''
5609  027756     340     340                  .BYTE   340,340                 ;PRIO. AT 7
5610  027760  012777  001300  151374          MOV     #BIT9+BIT7+BIT6,@DVSCR
5611  027766  012737  000200  177776          MOV     #200,PS ;SET CPU PRIO AND ENABLE VECT. ''A''
5612  027774  000240                          NOP                             ;WAIST
5613  027776  104007                          HLT     7                       ;DV FAILED TO INTERUPT
5614  030000  005077  151356          1$:     CLR     @DVSCR                  ;DSABLE DV11
5615  030004  104400                          SCOPE                           ;SCOPE THIS TEST
5616  030006  012716  030000          2$:     MOV     #1$,(SP)                ;SET FOR RETURN
5617  030012  000002                          RTI                             ;
5618
5619
5620                                          ;************************ TEST 121 ******************************
5621                                          ;*TEST THAT BIT15(NPR STATUS INTR) WILL
5622                                          ;*SET WHEN AN ENTRY IS MADE INTO THE
5623                                          ;*NPR STATUS REGISTER.
5624                                          ;*ALSO VERIFY THAT READING THE DVNSR CLEARS DVSCR BIT15.
5625                                          ;:*************************************************************************
5626
5627                                  ;  TEST 121
5628                                  ;--------------
5629  030014  012737  000121  001226  TST121: MOV     #121,TSTNO
5630  030022  012737  030116  001216          MOV     #TST122,NEXT
5631  030030  104412                          MSTCLR                          ;RESET DV11
5632  030032  005777  151324                  TST     @DVSCR                  ;MAKE SURE 15 =0
5633  030036  100001                          BPL     64$                     ;BR IF OK
5634  030040  104000                          HLT     0                       ;15 S/B=0
5635  030042  012777  000010  151312  64$:    MOV     #BIT3,@DVSCR            ;SET SOURCE SEC
```

```
5636  030050  012777  030367  151322          MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT2+BIT1+BIT0,@DVSFR
5637  030056  005000                          CLR     R0                      ;XFR ALU RESULT TO DVNSR
5638  030060  104415                          ROMCLK                          ;EXECUTE
5639  030062  005777  151274          1$:     TST     @DVSCR                  ;15=1?
5640  030066  100404                          BMI     2$                      ;BR IF YES
5641  030070  104414                          DELAY                           ;WASTE TIME
5642  030072  005200                          INC     R0                      ;DO DELAY AND WAIT
5643  030074  001372                          BNE     1$                      ;
5644  030076  104000                          HLT     0                       ;15 NEVER SET
5645  030100  005777  151276          2$:     TST     @DVNSR                  ;READ NSR
5646  030104  005777  151252                  TST     @DVSCR                  ;DID 15 CLEAR IN SCR?
5647  030110  100001                          BPL     3$                      ;BR IF YES
5648  030112  104000                          HLT     0                       ;DVSCR IS NOT CLEARED BY READING NSR
5649  030114  104400          3$:             SCOPE                           ;SCOPE TEST
5650
5651
5652                                          ;************************ TEST 122 *******************************
5653                                          ;*TEST TO WRITE PATTERNS THROUGH
5654                                          ;*THE NPR STATUS REGISTER.
5655                                          ;*BITS WRITTEN: 11,10,09,08,03,02,01,00
5656                                          ;*(WHEN BIT 15 OF DVSCR IS SET SO SHOULD BIT 15 OF DVNSR)
5657                                          ;:*************************************************************
5658
5659                                  ;  TEST 122
5660                                  ;--------------
5661  030116  012737  000122  001226  TST122: MOV     #122,TSTNO
5662  030124  012737  030316  001216          MOV     #TST123,NEXT
5663  030132  012737  030174  001220          MOV     #1$,LOCK
5664  030140  104412                          MSTCLR                          ;RESET DV11
5665  030142  012777  000010  151212          MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
5666  030150  005000                          CLR     R0                      ;ZERO LINE NUMBER POINTER
5667  030152  005037  001250                  CLR     TEMP2                   ;ZERO DATA
5668  030156  013703  001402                  MOV     DVNSR,R3                ;SET POINTER
5669  030162  013702  001400                  MOV     DVSFR,R2                ;
5670  030166  012737  100000  001246          MOV     #BIT15,TEMP1            ;SET CORRESPONDING BIT TO BE FOUND IN NSR.
5671  030174  012712  030267          1$:     MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT1+BIT0,(R2)
5672  030200  104415                          ROMCLK                          ;XFR RAM OUTPUT TO NSR
5673  030202  005777  151154                  TST     @DVSCR                  ;WAIT FOR
5674  030206  100375                          BPL     .-4                     ;
5675  030210  013705  001246                  MOV     TEMP1,R5                ;GET GOOD 15=1 DATA
5676  030214  011304                          MOV     (R3),R4                 ;READ NSR
5677  030216  020504                          CMP     R5,R4                   ;OK?
5678  030220  001401                          BEQ     2$                      ;BR IF GOOD
5679  030222  104013                          HLT     13                      ;DVNSR HAS WRONG DATA
5680  030224  104401          2$:             SCOP1                           ;LOCK ON DATA?
5681  030226  004537  031676                  PERFORM ,SETSCAN                ;UPDATE MSCANNER
5682  030232  000001                          1                               ;BY ONE
5683  030234  005237  001246                  INC     TEMP1                   ;UPDATE DATA
5684  030240  032700  000020                  BIT     #BIT4,R0                ;ALL DONE?
5685  030244  001753                          BEQ     1$                      ;BR IF NO.
5686  030246  042737  077777  001246          BIC     #^C<BIT15>,TEMP1        ;CLEAR ALL BUT ENTRY PRESENT
5687  030254  005000                          CLR     R0                      ;ZERO LINE POINTER
5688  030256  005001                          CLR     R1                      ;ZERO MSCANNER, POINTER
5689  030260  005237  001250                  INC     TEMP2                   ;UPDATE HIGH BYTE DATA
5690  030264  153737  001250  001247          BISB    TEMP2,TEMP1+1           ;PLACE IN GOOD LOCATION
5691  030272  013712  001250                  MOV     TEMP2,(R2)              ;RAM ADDRESS
```

```
5692  030276  052712  020000                  BIS     #RAM,(R2)       ;RAM READ
5693  030302  104415                          ROMCLK                  ;
5694  030304  032737  000020  001250          BIT     #BIT4,TEMP2     ;ALL DONE?
5695  030312  001730                          BEQ     1$              ;BR IF NO
5696  030314  104400                          SCOPE                   ;SCOPE TEST.
5697
5698                                          ;**********  FIRST PLANNED ATTEMPT *********
5699                                          ;**********       TO EXECUTE NPR.   *********
5700                                          ;*----------------------------------------
5701
5702                                          ;************************* TEST 123 *******************************
5703                                          ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
5704                                          ;*TEST THAT THE DV11 CAN 'READ'' FROM CORE LOCATION
5705                                          ;*VIA THE NPR LOGIC.
5706                                          ;*LOACATION 'NPRLOC'' WILL HAVE A BINARY COUNT PATTERN
5707                                          ;*PLACED INTO IT AND READ INTO THE DV11 AND XFERED
5708                                          ;*INTO THE DVRIC REGISTER.
5709                                          ;*NOTE: THIS TEST USES AN EVEN ADDRESS FOR THE NPR OPERATION
5710                                          ;:*************************************************************************
5711
5712                                  ;  TEST 123
5713                                  ;--------------
5714  030316  012737  000123  001226  TST123: MOV     #123,TSTNO
5715  030324  012737  030456  001216          MOV     #TST124,NEXT
5716  030332  012737  030414  001220          MOV     #1$,LOCK
5717  030340  104412                          MSTCLR                  ;RESET DV11
5718  030342  012777  000010  151012          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
5719  030350  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
5720  030354  013702  001400                  MOV     DVSFR,R2        ;
5721  030360  012777  033352  151010          MOV     #NPRLOC,@DVSRA
5722  030366  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
5723  030372  104415                          ROMCLK                  ;EXECUTE
5724  030374  012712  030263                  MOV     #XFR+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
5725  030400  104415                          ROMCLK                  ;DO XFR TO NPR ADD.
5726  030402  005037  033352                  CLR     NPRLOC          ;CLEAR NPR LOCATION
5727  030406  005005                          CLR     R5              ;CLEAR GOOD
5728  030410  005077  150762                  CLR     @DVSRA          ;CLEAR DATA PORT
5729  030414  012712  040000          1$:     MOV     #NPR,(R2)       ;DO THE NPR
5730  030420  104415                          ROMCLK                  ;***NOW***
5731  030422  012712  030226                  MOV     #XFR+BIT7+BIT4+BIT2+BIT1,(R2)
5732  030426  104415                          ROMCLK                  ;XFR NPR OUTPUT TO DVRIC
5733  030430  011304                          MOV     (R3),R4         ;READ RIC
5734  030432  013705  033352                  MOV     NPRLOC,R5 ;READ GOOD DATA
5735  030436  020504                          CMP     R5,R4           ;OK?
5736  030440  001401                          BEQ     2$              ;BR IF YES
5737  030442  104013                          HLT     13              ;NPR FUNCTION FAILED
5738  030444  104401                  2$:     SCOP1                   ;LOOP?
5739  030446  105237  033352                  INCB    NPRLOC ;UPDATE DATA
5740  030452  001360                          BNE     1$              ;BR IF MORE
5741  030454  104400                          SCOPE                   ;SCOPE TEST
5742
5743
5744
5745                                          ;************************* TEST 124 *******************************
5746                                          ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
5747                                          ;*TEST THAT THE DV11 CAN 'READ'' FROM CORE LOCATION
```

```
                                        ;*VIA THE NPR LOGIC.
                                        ;*LOACATION 'NPRLOC' WILL HAVE A BINARY COUNT PATTERN
                                        ;*PLACED INTO IT AND READ INTO THE DV11 AND XFERED
                                        ;*INTO THE DVRIC REGISTER.
                                        ;*NOTE: THIS TEST USES AN ODD ADDRESS FOR THE NPR OPERATION.
                                        ;**********************************************************************

                                ;  TEST 124
                                ;--------------
030456  012737  000124  001226  TST124: MOV     #124,TSTNO
030464  012737  030620  001216         MOV     #TST125,NEXT
030472  012737  030554  001220         MOV     #1$,LOCK
030500  104412                        MSTCLR                  ;RESET DV11
030502  012777  000010  150652         MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
030510  013703  001366                 MOV     DVRIC,R3        ;SET POINTERS
030514  013702  001400                 MOV     DVSFR,R2        ;
030520  012777  033353  150650         MOV     #NPRLOC+1,@DVSRA
030526  012712  020000                 MOV     #RAM,(R2)       ;RAM READ
030532  104415                        ROMCLK                  ;EXECUTE
030534  012712  030263                 MOV     #XFR+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
030540  104415                        ROMCLK                  ;DO XFR TO NPR ADD.
030542  005037  033352                 CLR     NPRLOC          ;CLEAR NPR LOCATION
030546  005005                        CLR     R5              ;CLEAR GOOD
030550  005077  150622                 CLR     @DVSRA          ;CLEAR DATA PORT
030554  012712  040000         1$:     MOV     #NPR,(R2)       ;DO THE NPR
030560  104415                        ROMCLK                  ;***NOW***
030562  012712  030226                 MOV     #XFR+BIT7+BIT4+BIT2+BIT1,(R2)
030566  104415                        ROMCLK                  ;XFR NPR OUTPUT TO DVRIC
030570  011304                        MOV     (R3),R4         ;READ RIC
030572  013705  033352                 MOV     NPRLOC,R5       ;GET GOOD DATA
030576  000305                        SWAB    R5              ;
030600  020504                        CMP     R5,R4           ;OK?
030602  001401                        BEQ     2$              ;BR IF YES
030604  104013                        HLT     13              ;NPR FUNCTION FAILED
030606  104401                2$:     SCOP1                   ;LOOP?
030610  105237  033353                INCB    NPRLOC+1 ;UPDATE DATA
030614  001357                        BNE     1$              ;BR IF MORE
030616  104400                        SCOPE                   ;SCOPE TEST


                                        ;************************* TEST 125 *******************************
                                        ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
                                        ;*TEST THAT THE DV11 CAN 'WRITTEN' INTO CORE LOCATION
                                        ;*VIA THE NPR LOGIC.
                                        ;*LOACATION 'NPRLOC' WILL HAVE A BINARY COUNT PATTERN
                                        ;*WRITTEN INTO IT BY THE DV11 NPR LOGIC.
                                        ;*NOTE: THIS TEST USES AN EVEN ADDRESS FOR THE NPR OPERATION
                                        ;**********************************************************************

                                ;  TEST 125
                                ;--------------
030620  012737  000125  001226  TST125: MOV     #125,TSTNO
030626  012737  031002  001216         MOV     #TST126,NEXT
030634  012737  030716  001220         MOV     #1$,LOCK
030642  104412                        MSTCLR                  ;RESET DV11
```

```
5804  030644  012777  000010  150510          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
5805  030652  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
5806  030656  013702  001400                  MOV     DVSFR,R2        ;
5807  030662  012777  033352  150506          MOV     #NPRLOC,@DVSRA
5808  030670  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
5809  030674  104415                          ROMCLK                  ;EXECUTE
5810  030676  012712  031663                  MOV     #XFR+BIT9+BIT8+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
5811  030702  104415                          ROMCLK                  ;DO XFR TO NPR ADD.
5812  030704  005037  033352                  CLR     NPRLOC          ;CLEAR NPR LOCATION
5813  030710  005005                          CLR     R5              ;CLEAR GOOD
5814  030712  005077  150460                  CLR     @DVSRA          ;CLEAR DATA PORT
5815  030716  005037  033352          1$:     CLR     NPRLOC          ;CLEAR NPR LOC
5816  030722  012712  020000                  MOV     #RAM,(R2)       ;DO RAM READ
5817  030726  104415                          ROMCLK                  ;
5818  030730  012712  030265                  MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT0,(R2)
5819  030734  104415                          ROMCLK                  ;XFR RAM OUTPUT TO NPR INPUT DATA
5820  030736  012712  040000                  MOV     #NPR,(R2)       ;DO THE NPR
5821  030742  104415                          ROMCLK                  ;***NOW***
5822  030744  017705  150426                  MOV     @DVSRA,R5       ;READ GOOD DATA
5823  030750  013704  033352                  MOV     NPRLOC,R4 ;READ ???? DATA
5824  030754  020504                          CMP     R5,R4           ;OK?
5825  030756  001401                          BEQ     2$              ;BR IF YES
5826  030760  104013                          HLT     13              ;NPR FUNCTION FAILED
5827  030762  104401                  2$:     SCOP1                   ;LOOP?
5828  030764  005277  150406                  INC     @DVSRA          ;UPDATE DATA
5829  030770  032777  000400  150400          BIT     #BIT8,@DVSRA    ;ALL DONE?
5830  030776  001747                          BEQ     1$              ;BR IF NO
5831  031000  104400                          SCOPE                   ;SCOPE TEST
5832
5833
5834
5835                                  ;************************* TEST 126 ******************************
5836                                  ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
5837                                  ;*TEST THAT THE DV11 CAN 'WRITTEN'' INTO CORE LOCATION
5838                                  ;*VIA THE NPR LOGIC.
5839                                  ;*LOACATION 'NPRLOC'' WILL HAVE A BINARY COUNT PATTERN
5840                                  ;*WRITTEN INTO IT BY THE DV11 NPR LOGIC.
5841                                  ;*NOTE: THIS TEST USES AN ODD ADDRESS FOR THE NPR OPERATION.
5842                                  ;;*****************************************************************************
5843
5844                          ;  TEST 126
5845                          ;--------------
5846  031002  012737  000126  001226  TST126: MOV     #126,TSTNO
5847  031010  012737  031166  001216          MOV     #TST127,NEXT
5848  031016  012737  031100  001220          MOV     #1$,LOCK
5849  031024  104412                          MSTCLR                  ;RESET DV11
5850  031026  012777  000010  150326          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
5851  031034  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
5852  031040  013702  001400                  MOV     DVSFR,R2        ;
5853  031044  012777  033353  150324          MOV     #NPRLOC+1,@DVSRA
5854  031052  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
5855  031056  104415                          ROMCLK                  ;EXECUTE
5856  031060  012712  031663                  MOV     #XFR+BIT9+BIT8+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
5857  031064  104415                          ROMCLK                  ;DO XFR TO NPR ADD.
5858  031066  005037  033352                  CLR     NPRLOC          ;CLEAR NPR LOCATION
5859  031072  005005                          CLR     R5              ;CLEAR GOOD
```

```
5860  031074  005077  150276            CLR     @DVSRA          ;CLEAR DATA PORT
5861  031100  005037  033352      1$:   CLR     NPRLOC          ;CLEAR NPR LOC
5862  031104  012712  020000            MOV     #RAM,(R2)       ;DO RAM READ
5863  031110  104415                    ROMCLK                  ;
5864  031112  012712  030265            MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT0,(R2)
5865  031116  104415                    ROMCLK                  ;XFR RAM OUTPUT TO NPR INPUT DATA
5866  031120  012712  040000            MOV     #NPR,(R2)       ;DO THE NPR
5867  031124  104415                    ROMCLK                  ;***NOW***
5868  031126  017705  150244            MOV     @DVSRA,R5       ;READ GOOD DATA
5869  031132  013704  033352            MOV     NPRLOC,R4       ;GET DATA
5870  031136  000304                    SWAB    R4              ;
5871  031140  020504                    CMP     R5,R4           ;OK?
5872  031142  001401                    BEQ     2$              ;BR IF YES
5873  031144  104013                    HLT     13              ;NPR FUNCTION FAILED
5874  031146  104401            2$:     SCOP1                   ;LOOP?
5875  031150  005277  150222            INC     @DVSRA          ;UPDATE DATA
5876  031154  032777  000400  150214    BIT     #BIT8,@DVSRA    ;ALL DONE?
5877  031162  001746                    BEQ     1$              ;BR IF NO
5878  031164  104400                    SCOPE                   ;SCOPE TEST
5879
5880
5881
5882                                    ;************************* TEST 127 ******************************
5883                                    ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
5884                                    ;*TEST THAT THE DV11 CAN DO AN NPR
5885                                    ;*TO A NON-EXISTANT MEMORY.
5886                                    ;*TEST THAT BRANCH "A" -NXM H- IS SET AFTER
5887                                    ;*THE NPR. THEN DO A "SET/CLEAR" CLEAR NXM
5888                                    ;*AND VERIFY THAT IT IS CLEARED.
5889                                    ;;*****************************************************************
5890
5891                                ;  TEST 127
5892                                ;--------------
5893  031166  012737  000127  001226 TST127: MOV     #127,TSTNO
5894  031174  012737  031344  001216         MOV     #TST130,NEXT
5895  031202  104412                        MSTCLR                  ;RESET DV11
5896  031204  012777  000010  150150        MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
5897  031212  013703  001366                MOV     DVRIC,R3        ;SET POINTERS
5898  031216  013702  001400                MOV     DVSFR,R2        ;
5899  031222  052777  000060  150132        BIS     #BIT5+BIT4,@DVSCR       ;SET EA BITS.
5900  031230  012777  177320  150140        MOV     #177320,@DVSRA  ;LOAD 'NXM'.
5901  031236  012712  020000                MOV     #RAM,(R2)       ;RAM READ
5902  031242  104415                        ROMCLK                  ;EXECUTE
5903  031244  012712  031663                MOV     #XFR+BIT9+BIT8+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
5904  031250  104415                        ROMCLK                  ;EXECUTE
5905  031252  012712  040000                MOV     #NPR,(R2)       ;DO THE NPR
5906  031256  104415                        ROMCLK                  ;***NOW***
5907  031260  012712  003000                MOV     #BIT10+BIT9,(R2)        ;BRANCH "A" NXM
5908  031264  017704  150100                MOV     @DVLCR,R4       ;READ BRANCH TEST POINTS.
5909  031270  010405                        MOV     R4,R5           ;GET IMAGE
5910  031272  042705  000001                BIC     #BIT0,R5        ;BR "A" TRUE
5911  031276  052705  000002                BIS     #BIT1,R5        ;BR B FALSE
5912  031302  020504                        CMP     R5,R4           ;NXM TRUE?
5913  031304  001401                        BEQ     .+4             ;BR IF YES
5914  031306  104006                        HLT     6               ;BR "A" OF NXM FAILED!
5915  031310  012712  050017                MOV     #S.C+BIT3+BIT2+BIT1+BIT0,(R2)
```

```
                                                        ;SET/CLEAR  CLEAR NXM.
031314  104415                          ROMCLK          ;EXECUTE.
031316  012712  003000                  MOV     #BIT10+BIT9,(R2)        ;BRANCH ''A'' NXM
031322  017704  150042                  MOV     @DVLCR,R4       ;READ BRANCH TEST POINTS.
031326  010405                          MOV     R4,R5           ;GET IMAGE
031330  052705  000003                  BIS     #BIT1+BIT0,R5   ;BR A FALSE  BR B FALSE
031334  020504                          CMP     R5,R4           ;NXM TRUE?
031336  001401                          BEQ     .+4             ;BR IF NO
031340  104006                          HLT     6               ;BR ''A'' OF NXM FAILED TO CLEAR!
031342  104400                          SCOPE                   ;SCOPE TEST

                                        ;************************ TEST 130 *********************************
                                        ;*TEST MEMORY EXT. BITS 16 AND 17 UTILIZING
                                        ;*THE ALU SECTION OF THE DV11 MICROPROCESSOR.
                                        ;*ENABLES LINE 0, E.A. BIT 17 = 1, BIT 16 = 0
                                        ;*SEC REG 0 -> REG ''A'' -> ALU -> SEC REG 2
                                        ;*SEC REG 0 = SEC REG 2?
                                        ;*SEC REG 0 -> REG ''B'' -> ALU -> SEC REG 4
                                        ;*SEC REG 0 = SEC REG 4?
                                        ;*REPEAT ABOVE STEPS FOR E.A.BIT 16 = 1, BIT 17 = 0
                                        ;:*******************************************************************

                                ;  TEST 130
                                ;--------------
031344  012737  000130  001226  TST130: MOV     #130,TSTNO
031352  012737  002436  001216          MOV     #.EOP,NEXT
031360  104412                          MSTCLR                  ;RESET DV11,DISABLE MICROPROCESSOR
031362  012701  014622                  MOV     #MEMEXT,R1      ;R1 IS SEC REG POINTER
031366  013700  001400                  MOV     DVSFR,R0        ;SET SP. FUNCTIONS REG POINTER
031372  012737  000002  001254          MOV     #2,TEMP4        ;ENABLE LOOP COUNTER
031400  012703  000003                  MOV     #3,R3           ;R3 IS SEC REG COUNTER
031404  005077  147762                  CLR     @DVSRS          ;ENABLE LINE 0
031410  111177  147760          1$:     MOVB    (R1),@DVSRSH    ;SELECT THE SEC REG
031414  005077  147756                  CLR     @DVSRA          ;CLEAR THE SEC REG
031420  005201                          INC     R1              ;ADD 1 TO SEC REG POINTER
031422  005303                          DEC     R3              ;ANOTHER SEC REG TO CLEAR?
031424  001371                          BNE     1$              ;YES
031426  012777  000050  147726          MOV     #BIT5+BIT3,@DVSCR
                                                        ;ENABLE E.A. BIT 17 & ROM DATA SOURCE
031434  012701  014622                  MOV     #MEMEXT,R1      ;RESTORE SEC REG POINTER
031440  004537  031640          2$:     JSR     R5,SCREG0       ;HAVE SUBROUTINE SET UP SEC REG 0
031444  012710  030261                  MOV     #XFR+BIT7+BIT5+BIT4+BIT0,(R0)
                                                        ;SEC REG 0 -> REG ''A''
031450  104415                          ROMCLK                  ;EXECUTE
031452  012710  010037                  MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,(R0)  ;F = A
031456  104415                          ROMCLK                  ;EXECUTE
031460  116177  000001  147706          MOVB    1(R1),@DVSRSH   ;SET FOR SEC REG 2
031466  012710  020002                  MOV     #RAM+BIT1,(R0)  ;CLEAR ''FOOTPRINT'' REG. FOR SEC. REG.2
031472  104415                          ROMCLK                  ;EXECUTE
031474  012710  020762                  MOV     #RAM+BIT8+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
                                                        ;ALU RESULT 0-17 -> SEC REG 2
031500  104415                          ROMCLK                  ;EXECUTE
031502  012710  020002                  MOV     #RAM+BIT1,(R0)  ;READ SEC REG 2
031506  104415                          ROMCLK                  ;EXECUTE
031510  017702  147654                  MOV     @DVLCR,R2       ;SAVE DVLINE PARM. REG
```

```
5972  031514  042702  177717              BIC     #^C<BIT5+BIT4>,R2
5973                                                ;CLEAR ALL EXCEPT E.A. BITS
5974  031520  120402                      CMPB    R4,R2           ;SEC REG 0 = SEC REG 2?
5975  031522  001041                      BNE     3$              ;NO,ERROR
5976  031524  004537  031640              JSR     R5,SCREG0       ;HAVE SUBROUTINE SET UP SEC REG 0
5977  031530  012710  030262              MOV     #XFR+BIT7+BIT5+BIT4+BIT1,(R0)
5978                                              ;SEC REG 0 -> REG 'B'
5979  031534  104415                      ROMCLK                  ;EXECUTE
5980  031536  012710  010005              MOV     #ALU+BIT2+BIT0,(R0)  ;F = B
5981  031542  104415                      ROMCLK                  ;EXECUTE
5982  031544  116177  000002  147622      MOVB    2(R1),@DVSRSH   ;SET TO LOAD SEC REG 4
5983  031552  012710  020004              MOV     #RAM+BIT2,(R0)  ;CLEAR 'FOOTPRINT' REG. FOR SEC. REG.4
5984  031556  104415                      ROMCLK                  ;EXECUTE
5985  031560  012710  020764              MOV     #RAM+BIT8+BIT7+BIT6+BIT5+BIT4+BIT2,(R0)
5986                                              ;ALU RESULT 0-17 -> SEC REG 4
5987  031564  104415                      ROMCLK                  ;EXECUTE
5988  031566  012710  020004              MOV     #RAM+BIT2,(R0)  ;READ SEC REG 4
5989  031572  104415                      ROMCLK                  ;EXECUTE
5990  031574  017702  147570              MOV     @DVLCR,R2       ;SAVE DVLINE PARM. REG
5991  031600  042702  177717              BIC     #^C<BIT5+BIT4>,R2  ;CLEAR ALL BUT E.A. BITS
5992  031604  120402                      CMPB    R4,R2           ;SEC REG 0 = SEC REG 4?
5993  031606  001007                      BNE     3$              ;NO,ERROR
5994  031610  005337  001254              DEC     TEMP4           ;1ST TIME THROUGH?
5995  031614  001405                      BEQ     4$              ;NO, FINISHED OK
5996  031616  012777  000030  147536      MOV     #BIT4+BIT3,@DVSCR
5997                                              ;ENABLE E.A. BIT 16 & ROM DATA SOURCE
5998  031624  000705                      BR      2$              ;PROCESS E.A. BIT 16
5999  031626  104015              3$:     HLT     15              ;ERROR MESSAGE
6000  031630  012777  000001  147524  4$: MOV     #1,@DVSCR       ;SET MICROPROCESSOR GO
6001  031636  104400                      SCOPE                   ;SCOPE TEST
6002
6003                                      ;SUBROUTINE TO SET UP SEC REG 0 FOR
6004                                      ;MEMORY EXTENSION BITS/ALU TEST
6005  031640  005077  147526      SCREG0: CLR     @DVSRS          ;ENABLE LINE 0
6006  031644  111177  147524              MOVB    (R1),@DVSRSH    ;SELECT SEC REG 0
6007  031650  112777  000100  147520      MOVB    #100,@DVSRA     ;SET LOW BYTE TRACE DATA
6008  031656  012710  020000              MOV     #RAM,(R0)       ;READ SEC REG 0
6009  031662  104415                      ROMCLK                  ;EXECUTE
6010  031664  117704  147500              MOVB    @DVLCR,R4       ;SAVE DVLINE PARM REG
6011  031670  042704  177717              BIC     #^C<BIT5+BIT4>,R4
6012                                              ;CLEAR ALL BUT E.A. BITS
6013  031674  000205                      RTS     R5              ;RETURN TO MAIN PROGRAM
6014
```

```
6015
6016  031676                                SETSCAN:
6017  031676  010346                              MOV     R3,-(SP)
6018  031700  052777  000010  147454              BIS     #BIT3,@DVSCR
6019  031706  012503                              MOV     (R5)+,R3
6020  031710  001414                              BEQ     2$
6021  031712  012777  050102  147460  1$:         MOV     #BIT14+BIT12+BIT6+BIT1,@DVSFR
6022  031720  104415                              ROMCLK
6023  031722  005201                              INC     R1
6024  031724  012777  050102  147446              MOV     #BIT14+BIT12+BIT6+BIT1,@DVSFR
6025  031732  104415                              ROMCLK
6026  031734  005201                              INC     R1
6027  031736  005303                              DEC     R3
6028  031740  001364                              BNE     1$
6029  031742  012603                    2$:       MOV     (SP)+,R3
6030  031744  010100                              MOV     R1,R0
6031  031746  000241                              CLC
6032  031750  006000                              ROR     R0
6033  031752  000205                              EXIT
6034
6035                                              ;SUBROUTINE TO LOAD DATA INTO ''A'' AND 'B'' REG.
6036                                              ;THE FIRST ARGUMENT WILL LOAD THE ''A'' REGISTER;
6037                                              ;THE SECOND ARGUMENT WILL LOAD THE 'B'' REGISTER;
6038                                              ;AND THE THIRD ARGUMENT WILL SPECIFY THE POLYNOMIAL USED.
6039
6040  031754  012377  147416            L.DATA:   MOV     (R3)+,@DVSRA    ;LOAD DATA TO BE PLACED INTO THE ''A'' REG
6041  031760  012710  020000                      MOV     #RAM,(R0)       ;DO A ROM READ INSTR.
6042  031764  104415                              ROMCLK                  ;EXECUTE
6043  031766  012710  030261                      MOV     #XFR+BIT7+BIT5+BIT4+BIT0,(R0)
6044  031772  104415                              ROMCLK                  ;DATAXFR FROM RAM OUTPUT TO ''A'' REGISTER
6045  031774  012377  147376                      MOV     (R3)+,@DVSRA    ;LOAD DATA TO BE PLACED INTO THE 'B'' REG.
6046  032000  012710  020000                      MOV     #RAM,(R0)       ;DO A ROM READ
6047  032004  104415                              ROMCLK                  ;
6048  032006  012710  030262                      MOV     #XFR+BIT7+BIT5+BIT4+BIT1,(R0)
6049  032012  104415                              ROMCLK                  ;DO A DATA XFER FROM RAM OUTPUT TO THE 'B'' REG.
6050  032014  012377  147356                      MOV     (R3)+,@DVSRA    ;PLACE DATA TO REMAIN IN THE RAMOUTPUT REG
6051  032020  012710  020000                      MOV     #RAM,(R0)       ;DO A RAM READ TO SPECIFY POLYN.
6052  032024  104415                              ROMCLK                  ;READ
6053  032026  000203                              RTS     R3              ;LEAVE HERE
6054
6055
6056  032030  010046                    SIMBCC:   MOV     R0,-(SP)
6057  032032  010146                              MOV     R1,-(SP)
6058  032034  010246                              MOV     R2,-(SP)
6059  032036  012537  001246                      MOV     (R5)+,TEMP1
6060  032042  012537  001250                      MOV     (R5)+,TEMP2
6061  032046  012537  001252                      MOV     (R5)+,TEMP3
6062  032052  005037  032204            1$:       CLR     BCCFBK
6063  032056  013700  001252                      MOV     TEMP3,R0
6064  032062  006037  001250                      ROR     TEMP2
6065  032066  005500                              ADC     R0
6066  032070  032700  000001                      BIT     #BIT0,R0
6067  032074  001402                              BEQ     2$
6068  032076  005137  032204                      COM     BCCFBK
6069  032102  013700  032202            2$:       MOV     XPOLY,R0
6070  032106  005100                              COM     R0
```

```
6071  032110  040037  032204          BIC     R0,BCCFBK
6072  032114  000241                  CLC
6073  032116  006037  001252          ROR     TEMP3
6074  032122  013700  032204          MOV     BCCFBK,R0
6075  032126  013701  001252          MOV     TEMP3,R1
6076  032132  010102                  MOV     R1,R2
6077  032134  040100                  BIC     R1,R0
6078  032136  043702  032204          BIC     BCCFBK,R2
6079  032142  050200                  BIS     R2,R0
6080  032144  043737  032202  001252  BIC     XPOLY,TEMP3
6081  032152  050037  001252          BIS     R0,TEMP3
6082  032156  005337  001246          DEC     TEMP1
6083  032162  001333                  BNE     1$
6084  032164  013737  001252  032206  MOV     TEMP3,CALBCC
6085  032172  012602                  MOV     (SP)+,R2
6086  032174  012601                  MOV     (SP)+,R1
6087  032176  012600                  MOV     (SP)+,R0
6088  032200  000205                  RTS     R5
6089  032202  000000          XPOLY:  0
6090  032204  000000          BCCFBK: 0
6091  032206  000000          CALBCC: 0
6092          000200          LRC8=200
6093          120001          CRC16=120001
6094          102010          CRC.CCITT=102010
6095
6096
6097  032210                  SETVEC:
6098  032210  012577  147136          MOV     (R5)+,@DVRVEC
6099  032214  012577  147136          MOV     (R5)+,@DVTVEC
6100  032220  112577  147130          MOVB    (R5)+,@DVRLVL
6101  032224  112577  147130          MOVB    (R5)+,@DVTLVL
6102  032230  000205                  RTS     R5
6103
6104  032232                  NO.ATRAP:
6105  032232  104011                  HLT     11
6106  032234  000002                  RTI
6107
6108  032236                  NO.BTRAP:
6109  032236  104012                  HLT     12
6110  032240  000002                  RTI
6111
6112
```

```
 6113                                   .NLIST  BEX
      032242  050377  044522  040515  EM1:    .ASCIZ  <377>/PRIMARY REGISTER ADDRESSING TIME-OUT/
      032310  051777  041505  047117  EM2:    .ASCIZ  <377>^SECONDARY REGISTER READ/WRITE TEST^
      032354  050377  044522  040515  EM3:    .ASCIZ  <377>^PRIMARY REGISTER READ/WRITE TEST^
      032416  046777  046505  051117  EM4:    .ASCIZ  <377>^MEMORY EXTENSION READ/WRITE TEST^
      032460  051777  042520  044503  EM5:    .ASCIZ  <377>/SPECIAL FUNCTION REG TEST/
      032513     377  042526  052103  EM6:    .ASCIZ  <377>/VECTOR "A" FAILED TO INTERUPT/
      032552  053377  041505  047524  EM7:    .ASCIZ  <377>/VECTOR "B" FAILED TO INTERUPT/
      032611     377  047125  054105  EM8:    .ASCIZ  <377>/UNEXPECTED INTERUPT ON VECTOR "A"/
      032654  052777  042516  050130  EM9:    .ASCIZ  <377>/UNEXPECTED INTERUPT ON VECTOR "B"/
      032717     377  051120  046511  EM10:   .ASCIZ  <377>/PRIMARY REGISTER ERROR/
      032747     377  044514  042516  EM11:   .ASCIZ  <377>/LINE CARD STATIC TEST/
      032776  046777  046505  051117  EM12:   .ASCIZ  <377>^MEMORY EXTENSION BITS/ALU TEST^
      033036  051377  043505  051511  DH1:    .ASCIZ  <377>/REGISTER REFERENCED  TRAPPED FROM/
      033101     377  054105  042520  DH2:    .ASCIZ  <377>/EXPECTED  FOUND   LINE  SEC REG  PRI REG/
      033153     377  054105  042520  DH3:    .ASCIZ  <377>/EXPECTED  FOUND   PRI REG/
      033206  045377  051123  050040  DH4:    .ASCIZ  <377>/JSR PC  DVSFR   EXPECTED  FOUND/
      033247     377  053104  043123  DH5:    .ASCIZ  <377>/DVSFR   EXPECTED  FOUND/
      033300  046777  052123  041523  DH6:    .ASCIZ  <377>/MSTSCAN  DVSFR  EXPECTED  FOUND  LINE/
              033350                          .EVEN
              000000                  SKIP=000000
 6114 033350  000000                  DATA:   0
 6115 033352  000000                  NPRLOC: 0
 6116 033354  000002                  DT1:    2
 6117 033356     006  017                      .BYTE   6,15.
 6118 033360  001262                          SAVR1
 6119 033362     006  002                      .BYTE   6,2
 6120 033364  001264                          SAVR2
 6121 033366  000005                  DT2:    5
 6122 033370     006  004                      .BYTE   6,4
 6123 033372  001270                          SAVR4
 6124 033374     006  002                      .BYTE   6,2
 6125 033376  001266                          SAVR3
 6126 033400     002  004                      .BYTE   2,4
 6127 033402  001260                          SAVR0
 6128 033404     002  007                      .BYTE   2,7
 6129 033406  001262                          SAVR1
 6130 033410     006  002                      .BYTE   6,2
 6131 033412  001264                          SAVR2
 6132 033414  000003                  DT3:    3
 6133 033416     006  004                      .BYTE   6,4
 6134 033420  001272                          SAVR5
 6135 033422     006  002                      .BYTE   6,2
 6136 033424  001270                          SAVR4
 6137 033426     006  002                      .BYTE   6,2
 6138 033430  001266                          SAVR3
 6139 033432  000004                  DT4:    4
 6140 033434     006  002                      .BYTE   6,2
 6141 033436  001262                          SAVR1
 6142 033440     006  002                      .BYTE   6,2
 6143 033442  001266                          SAVR3
 6144 033444     006  004                      .BYTE   6,4
 6145 033446  001272                          SAVR5
 6146 033450     006  001                      .BYTE   6,1
 6147 033452  001270                          SAVR4
 6148 033454                          DT5:
```

```
 6149  033454  000003                          3
 6150  033456     006     002                  .BYTE   6,2
 6151  033460  001264                          SAVR2
 6152  033462     006     004                  .BYTE   6,4
 6153  033464  001272                          SAVR5
 6154  033466     006     001                  .BYTE   6,1
 6155  033470  001270                          SAVR4
 6156  033472  000005                  DT6:    5
 6157  033474     006     003                  .BYTE   6,3
 6158  033476  001260                          SAVR0
 6159  033500     006     001                  .BYTE   6,1
 6160  033502  001264                          SAVR2
 6161  033504     006     004                  .BYTE   6,4
 6162  033506  001272                          SAVR5
 6163  033510     006     001                  .BYTE   6,1
 6164  033512  001270                          SAVR4
 6165  033514     002     001                  .BYTE   2,1
 6166  033516  001262                          SAVR1
 6167
 6168  033520                          .ERRTAB:
 6169  033520  000000                          0
 6170  033522  000000                          0
 6171  033524  000000                          0
 6172  033526  032242                          EM1
 6173  033530  033036                          DH1     ;HALT 1
 6174  033532  033354                          DT1
 6175  033534  032310                          EM2
 6176  033536  033101                          DH2     ;HALT 2
 6177  033540  033366                          DT2
 6178  033542  032354                          EM3
 6179  033544  033153                          DH3     ;HALT 3
 6180  033546  033414                          DT3
 6181  033550  032416                          EM4
 6182  033552  033101                          DH2     ;HALT 4
 6183  033554  033366                          DT2
 6184  033556  032460                          EM5
 6185  033560  033206                          DH4     ;HALT 5
 6186  033562  033432                          DT4
 6187  033564  032460                          EM5
 6188  033566  033247                          DH5     ;HALT 6
 6189  033570  033454                          DT5
 6190  033572  032513                          EM6
 6191  033574  000000                          0       ;HALT   7
 6192  033576  000000                          0
 6193  033600  032552                          EM7
 6194  033602  000000                          0       ;HALT   10
 6195  033604  000000                          0
 6196  033606  032611                          EM8
 6197  033610  000000                          0       ;HALT   11
 6198  033612  000000                          0
 6199  033614  032654                          EM9
 6200  033616  000000                          0       ;HALT 12
 6201  033620  000000                          0
 6202  033622  032717                          EM10
 6203  033624  033153                          DH3     ;HALT 13
 6204  033626  033414                          DT3
```

```
 6205  033630  032747                          EM11
 6206  033632  033300                          DH6     ;HALT   14
 6207  033634  033472                          DT6
 6208  033636  032776                          EM12
 6209  033640  033101                          DH2     ;HALT 15
 6210  033642  033366                          DT2
 6211  033644                          CORMAX:
 6212          000001                  .END
```

```
ADRCNT= 003443   1157* 1193* 1202#
ALU   = 010000    611# 4106 4113 4148 4171 4173 4187 4197 4204 4224 4230 4252 4256
                  4258 4275 4294 4439 4467 4495 4523 4551 4579 4607 4635 4722 4846
                  4861 4876 4929 5961 5980
ASYNC = 004000    619#
AUTO.S  006624   1667#
BCC   = 060000    616# 4838 4850 4865 4882 4916 4967 4989 5011 5033 5055 5077 5099
                  5121 5143 5165 5187 5209 5253 5298 5343
BCCFBK  032204   6062* 6068* 6071* 6074 6078 6090#
BINWRD  003746   1243* 1244 1281#
BIT0  = 000001    609# 1424 2153 2258 2264 2286 2292 2314 2320 2342 2348 2370 2376
                  2398 2404 2423 2732 2952 2967 3140 3145 3164 3279 3344 3447 3456
                  3463 3465 3472 3474 3481 3483 3490 3492 3514 3541 3543 3550 3552
                  3559 3561 3568 3570 3608 3663 3687 3721 3723 3730 3732 3734 3736
                  3743 3745 3747 3749 3758 3762 3771 3775 3782 3784 3786 3788 3795
                  3797 3799 3801 3810 3814 3823 3827 3859 3860 3871 3879 3889 3895
                  3916 3923 3932 3945 3951 4020 4021 4042 4043 4106 4187 4195 4204
                  4224 4226 4258 4296 4437 4439 4443 4452 4453 4465 4467 4471 4480
                  4481 4493 4495 4499 4508 4509 4518 4521 4523 4527 4536 4537 4549
                  4551 4555 4564 4565 4577 4579 4583 4592 4593 4605 4607 4611 4620
                  4621 4633 4635 4639 4648 4649 4676 4677 4681 4687 4697 4700 4722
                  4726 4741 4743 4767 4769 4863 4878 4908 5636 5671 5724 5767 5810
                  5818 5856 5864 5903 5910 5915 5921 5958 5961 5980 6043 6066
BIT1  = 000002    608# 1424 1435 2258 2264 2286 2292 2314 2320 2342 2348 2370 2376
                  2398 2404 2449 2758 3140 3164 3279 3344 3413 3445 3447 3454 3456
                  3463 3465 3472 3474 3481 3483 3490 3492 3514 3541 3543 3550 3552
                  3559 3561 3568 3570 3597 3608 3618 3719 3723 3730 3734 3736 3749
                  3756 3760 3762 3775 3782 3786 3788 3801 3808 3812 3814 3827 3859
                  3916 3931 3945 3998 4018 4020 4021 4029 4040 4042 4043 4051 4099
                  4106 4113 4147 4187 4188 4190 4196 4198 4200 4204 4205 4207 4224
                  4228 4229 4230 4231 4233 4254 4277 4296 4324 4333 4335 4383 4439
                  4443 4444 4452 4453 4467 4471 4472 4480 4481 4495 4499 4500 4508
                  4509 4523 4527 4528 4536 4537 4546 4551 4555 4556 4564 4565 4579
                  4583 4584 4592 4593 4607 4611 4612 4620 4621 4635 4639 4640 4648
                  4649 4677 4681 4687 4690 4700 4722 4743 4767 4769 4837 4848 4880
                  4910 4918 4929 4931 4953 5235 5280 5325 5636 5671 5724 5731 5767
                  5774 5810 5856 5903 5911 5915 5921 5961 5964 5966 5969 5977 6021
                  6024 6048
BIT10 = 002000    599# 1424 2075 2283 2579 3140 3164 3452 3470 3479 3488 3511 3566
                  3617 3718 3722 3783 3787 3796 3800 3809 3813 3822 3826 3862 3875
                  3886 4581 4590 4609 4618 4637 4646 4672 4737 4762 5415 5422 5907
                  5918
BIT11 = 004000    598# 1424 2311 2605 3140 3164 3259 3325 3479 3511 3539 3548 3557
                  3566 3718 3722 3862 3875 3886 4469 4478 4497 4506 4525 4534 4553
                  4562 4581 4590 4609 4618 4637 4646 4672 4698 4737 4762
BIT12 = 010000    597#  611  613  615  617 2101 2339 3686 4490 5413 5422 6021 6024
BIT13 = 020000    596#  612  613  616  617 2127 2367 3674 4462 5455 5463 5498
BIT14 = 040000    595#  614  615  616  617 1064 2395 3662 4934 4984 4987 6021 6024
BIT15 = 100000    594# 2182 2186 2192 2193 3650 4434 5453 5463 5498 5670 5686
BIT2  = 000004    607#  991 1424 1860 2475 2784 3140 3164 3597 3675 3687 3717 3721
                  3730 3743 3756 3769 3782 3795 3808 3821 3998 4018 4020 4029 4031
                  4040 4042 4051 4053 4099 4106 4113 4147 4148 4171 4173 4187 4197
                  4204 4224 4230 4252 4256 4258 4275 4277 4294 4296 4324 4383 4403
                  4439 4467 4495 4523 4551 4574 4579 4607 4635 4722 4785 4837 4846
                  4861 4876 4918 4929 4953 5235 5280 5325 5636 5671 5731 5774 5818
                  5864 5915 5961 5980 5983 5985 5988
```

```
BIT3  = 000010   606# 1424 1886 2153 2501 2668 2810 2945 2951 2960 2966 3140 3164
                3404 3410 3442 3510 3538 3595 3597 3598 3608 3618 3649 3651 3661
                3663 3673 3675 3685 3687 3711 3858 3915 3972 4027 4097 4106 4145
                4148 4170 4171 4173 4187 4197 4204 4220 4224 4251 4252 4256 4274
                4275 4293 4294 4322 4380 4433 4439 4449 4461 4467 4477 4489 4495
                4505 4517 4523 4533 4545 4551 4561 4573 4579 4589 4601 4602 4607
                4617 4629 4635 4645 4673 4720 4722 4835 4846 4860 4861 4875 4876
                4903 4912 4957 4979 5001 5023 5045 5067 5089 5111 5133 5155 5177
                5199 5240 5285 5330 5635 5665 5718 5761 5804 5850 5896 5915 5954
                5961 5996 6018
BIT4  = 000020   605# 1912 2258 2264 2286 2292 2314 2320 2342 2348 2370 2376 2398
                2404 2836 3206 3217 3228 3279 3344 3717 3721 3860 3871 3883 3923
                4000 4007 4010 4029 4038 4040 4051 4099 4106 4113 4147 4148 4171
                4187 4195 4204 4224 4226 4228 4230 4252 4254 4275 4277 4294 4296
                4324 4383 4403 4437 4439 4465 4467 4493 4495 4521 4523 4549 4551
                4577 4579 4605 4607 4630 4633 4635 4722 4724 4726 4785 4846 4848
                4861 4863 4876 4878 4880 4908 4910 4929 4931 5636 5671 5684 5694
                5724 5731 5767 5774 5810 5818 5856 5864 5899 5903 5958 5961 5966
                5972 5977 5985 5991 5996 6011 6043 6048
BIT5  = 000040   604# 1938 2258 2264 2286 2292 2314 2320 2342 2348 2370 2376 2398
                2404 2862 3206 3217 3279 3344 3730 3734 3743 3747 3756 3760 3769
                3773 4000 4010 4018 4020 4021 4029 4031 4032 4040 4051 4099 4147
                4187 4195 4224 4226 4228 4254 4383 4437 4465 4493 4521 4549 4577
                4605 4633 4722 4724 4726 4837 4848 4863 4878 4880 4908 4910 4918
                4931 4953 5235 5280 5325 5636 5671 5724 5767 5810 5818 5856 5864
                5899 5903 5954 5958 5966 5972 5977 5985 5991 6011 6043 6048
BIT6  = 000100   603# 1735 1964 2888 4000 4010 4099 4147 4195 4226 4228 4254 4324
                4333 4335 4403 4724 4726 4785 4837 4848 4863 4878 4880 4918 4931
                4953 5235 5280 5325 5375 5382 5499 5511 5535 5560 5585 5610 5636
                5966 5985 6021 6024
BIT7  = 000200   602# 1058 1305 1462 1483 1735 1993 1997 2003 2004 2211 2914 3607
                3782 3786 3795 3799 3808 3812 3821 3825 3946 3952 4000 4010 4018
                4029 4040 4042 4043 4049 4051 4053 4054 4099 4147 4195 4226 4228
                4254 4383 4437 4465 4493 4521 4549 4577 4605 4633 4724 4726 4837
                4848 4863 4878 4880 4908 4910 4918 4931 4953 5235 5280 5325 5373
                5382 5499 5511 5535 5560 5585 5610 5636 5671 5724 5731 5767 5774
                5810 5818 5856 5864 5903 5958 5966 5977 5985 6043 6048
BIT8  = 000400   601# 1424 1441 2023 2210 2212 2527 2951 2966 3140 3164 3442 3443
                3452 3461 3470 3479 3511 3548 3594 3744 3748 3770 3774 3796 3800
                3822 3826 3862 3875 3886 3917 3928 4016 4553 4562 4609 4618 4698
                4724 5810 5829 5856 5876 5903 5966 5985
BIT9  = 001000   600# 1424 1735 1740 1997 2049 2186 2255 2553 3140 3164 3470 3479
                3488 3511 3557 3566 3757 3761 3770 3774 3809 3813 3822 3826 3862
                3875 3886 3917 3928 3946 4525 4534 4553 4562 4637 4646 4698 4737
                4762 4905 4915 4933 5373 5375 5382 5453 5455 5458 5463 5497 5499
                5535 5560 5585 5610 5810 5856 5903 5907 5918
BRB   = 070000   617# 3511 3539 3548 3557 3566 3731 3735 3744 3748 3757 3761 3770
                3774 3783 3787 3796 3800 3809 3813 3822 3826 4672 4698 4737 4762
BRW     003014   997 1086#
BRX     003016   998 1087#
CALBCC  032206  4958* 4971 4980* 4993 5002* 5015 5024* 5037 5046* 5059 5068* 5081 5090*
                5103 5112* 5125 5134* 5147 5156* 5169 5178* 5191 5200* 5213 5238* 5243
                5244 5257 5283* 5288 5289 5302 5328* 5333 5334 5347 6084* 6091#
CHRCNT  003744  1241* 1245 1261* 1279# 1280
CLKX    001242   688#
CLK.A   001416   788# 1585
```

```
CLK.B   001417            789#  1590
CLK.C   001420            790#  1595
CLK.D   001421            791#  1600
CNVRT = 104411            747#  1016  1018  1020  1022  1342  1344  1400  1457
CONVRT= 104410            745#   956  1358
CORMAX  033644           6211#  6212
CRC.CC= 102010           5324   6094#
CRC16 = 120001           4951   5279  6093#
CREAM   001306            709#   925* 1532* 1533  1535* 1540  1541* 1542  1545*
CSRMAP  006626            950   1669#
CYCLE   005666           1000   1036  1037  1522#
DATA    033350           6114#
DATABP  004276           1331*  1334  1356  1359#
DATACL= 104416            757#
DATAHD  004264           1330*  1352  1355#
DELAY = 104414            753#  5641
DEVADR  003440           1155*  1190  1200#
DH1     033036           6113#  6173
DH2     033101           6113#  6176  6182  6209
DH3     033153           6113#  6179  6203
DH4     033206           6113#  6185
DH5     033247           6113#  6188
DH6     033300           6113#  6206
DT1     033354           6116#  6174
DT2     033366           6121#  6177  6183  6210
DT3     033414           6132#  6180  6204
DT4     033432           6139#  6186
DT5     033454           6148#  6189
DT6     033472           6156#  6207
DVACTV  001300            703#   972*  973  1522  1527  1701* 1707* 1708* 1712  1731
DVCR00  001500            821#
DVCR01  001524            832#
DVCR02  001550            843#
DVCR03  001574            854#
DVCR04  001620            865#
DVCR05  001644            876#
DVCR06  001670            887#
DVCR07  001714            898#
DVLCR   001370            771#  1441* 1442  1561* 1562* 1563  1810  1832  2254  2282  2310  2338  2366
                         2394   3205  3216  3441  3515  3537  3837  3865  3878  3889  3894  3919  3932
                         3936   3947  3953  4442  4451  4470  4479  4498  4507  4526  4535  4554  4563
                         4582   4591  4610  4619  4638  4647  4675  4686  4699  4742  4768  5908  5919
                         5971   5990  6010
DVNSR   001402            776#  1571* 1572* 1573  1806  1828  2708  5645  5668
DVNUM   001301            704#   921  1026  1673* 1694* 1695  1702  1704
DVRIC   001366            770#  1559* 1560* 1561  2213  2233  3655  3667  3679  3691  4004  4062  4072
                         4103   4110  4117  4152  4164  4178  4191  4201  4208  4222  4242  4262  4328
                         4339   4388  4843  4855  4869  4887  4920  4972  4994  5016  5038  5060  5082
                         5104   5126  5148  5170  5192  5214  5258  5303  5348  5719  5762  5805  5851
                         5897
DVRLVL  001354            765#  1576* 1577* 1578  6100*
DVRVEC  001352            764#  1043  1547* 1576  6098*
DVSCR   001362            768#  1040  1418* 1431* 1435* 1546* 1557  1762  1793  1818  1859  1885  1911
                         1937   1963  1991  2022  2048  2074  2100  2126  2152  2180  2209  2668* 2943
                         3211*  3256  3266  3322  3330* 3331  3404* 3410* 3413  3442* 3461* 3510* 3538*
                         3594*  3595* 3601  3611  3622  3649* 3661* 3673* 3685* 3711* 3858* 3915* 3946*
```

```
                    3952* 3972* 4097* 4145* 4170* 4188* 4190* 4196* 4198* 4200* 4205* 4207* 4220*
                    4223  4251* 4274* 4293* 4322* 4380* 4433* 4449* 4461* 4477* 4489* 4505* 4517*
                    4533* 4545* 4561* 4573* 4589* 4601* 4617* 4629* 4645* 4673* 4720* 4835* 4860*
                    4875* 4903* 4957* 4979* 5001* 5023* 5045* 5067* 5089* 5111* 5133* 5155* 5177*
                    5199* 5240* 5285* 5330* 5373* 5375* 5377* 5382* 5389* 5392* 5413* 5415* 5417*
                    5422* 5429* 5432* 5453* 5455* 5457* 5458* 5463* 5470* 5473* 5497* 5498* 5499*
                    5511* 5535* 5538* 5560* 5563* 5585* 5588* 5610* 5614* 5632  5635* 5639  5646
                    5665* 5673  5718* 5761* 5804* 5850* 5896* 5899* 5954* 5996* 6000* 6018*
DVSCRH  001364       769# 1557* 1558* 1559
DVSFR   001400       775# 1569* 1570* 1571  2669  3406* 3407  3416  3440  3511* 3513  3536  3596
                    3648  3712  3860* 3862* 3864  3871* 3875* 3877  3883* 3886* 3893  3918* 3925*
                    3930* 3973  4098  4146  4323  4382  4435* 4437* 4439* 4441* 4450* 4463* 4465*
                    4467* 4469* 4478* 4491* 4493* 4495* 4497* 4506* 4519* 4521* 4523* 4525* 4534*
                    4547* 4549* 4551* 4553* 4562* 4575* 4577* 4579* 4581* 4590* 4603* 4605* 4607*
                    4609* 4618* 4631* 4633* 4635* 4637* 4646* 4674* 4683* 4685* 4698* 4703  4721
                    4737* 4762* 4766  4836  4904  4952  5236  5281  5326  5636* 5669  5720  5763
                    5806  5852  5898  5945  6021* 6024*
DVSRA   001376       774# 1422  1567* 1568* 1569  1796* 2631  2988  3048  3092  3136  3215* 3258*
                    3324* 4366  4434* 4462* 4490* 4518* 4546* 4574* 4602* 4630* 4682* 4732* 4747
                    4794  4905* 4912* 5721* 5728* 5764* 5771* 5807* 5814* 5822  5828* 5829  5853*
                    5860* 5868  5875* 5876  5900* 5950* 6007* 6040* 6045* 6050*
DVSRS   001372       772# 1421  1563* 1564* 1565  1795* 2422  2448  2474  2500  2526  2552  2578
                    2604  2958  2989* 3049* 3093* 3109* 3137* 3140* 3146* 3149* 3155  3164* 3173*
                    3212* 3257* 3323* 4364  4378* 4429* 4730* 4792  5948* 6005*
DVSRSH  001374       773# 1565* 1566* 1567  2990* 3050* 3060* 3094* 3110* 3147* 3158  3174* 3214*
                    4365  4731* 4793  5949* 5963* 5982* 6006*
DVTLVL  001360       767# 1580* 1581* 6101*
DVTR00  001502       822#
DVTR01  001526       833#
DVTR02  001552       844#
DVTR03  001576       855#
DVTR04  001622       866#
DVTR05  001646       877#
DVTR06  001672       888#
DVTR07  001716       899#
DVTVEC  001356       766# 1578* 1579* 1580  6099*
DV.END  001740       909# 1533  1542  1671
DV.MAP  001500       709   820#  925   952  1535  1545  1669  1674  1723
DV00.A  001504       823#
DV00.B  001510       825#
DV00.C  001514       827#
DV00.D  001520       829#
DV01.A  001530       834#
DV01.B  001534       836#
DV01.C  001540       838#
DV01.D  001544       840#
DV02.A  001554       845#
DV02.B  001560       847#
DV02.C  001564       849#
DV02.D  001570       851#
DV03.A  001600       856#
DV03.B  001604       858#
DV03.C  001610       860#
DV03.D  001614       862#
DV04.A  001624       867#
DV04.B  001630       869#
```

```
DV04.C  001634            871#
DV04.D  001640            873#
DV05.A  001650            878#
DV05.B  001654            880#
DV05.C  001660            882#
DV05.D  001664            884#
DV06.A  001674            889#
DV06.B  001700            891#
DV06.C  001704            893#
DV06.D  001710            895#
DV07.A  001720            900#
DV07.B  001724            902#
DV07.C  001730            904#
DV07.D  001734            906#
EM1     032242           6113#   6172
EM10    032717           6113#   6202
EM11    032747           6113#   6205
EM12    032776           6113#   6208
EM2     032310           6113#   6175
EM3     032354           6113#   6178
EM4     032416           6113#   6181
EM5     032460           6113#   6184    6187
EM6     032513           6113#   6190
EM7     032552           6113#   6193
EM8     032611           6113#   6196
EM9     032654           6113#   6199
ERRCNT  001232            680#    927*   1049    1367*
ERRFLG  001311            715#    923*   1011*   1078*   1319*   1332    1346*   1401*
ERRMSG  004252           1329*   1347    1350#
ERTABO  004366           1344    1376#
EXIT  = 000205            619#   6033
EXITER  004322           1362    1367#
FIX.00  006516           1586    1591    1596    1601    1635#
HALTS   004302           1315    1361#
HILIM   003436           1154*   1181    1199#
ICOUNT  001222            676#   1076    1081*   3086*
INBUF   005520           1124    1160    1507#
INIFLG  001310            714#    932     947*
INSTER= 104404            737#   1175
INSTR = 104403            735#   1608
INSTR2  003236           1131    1143#
LIGHT   000174            648#    942
LIGHTS  001200            659#    942*   1013*
LIMITS  003364           1170    1181#
LOBITS  003442           1156*   1185    1201#   1202
LOCK    001220            675#   1080*   1094    1096    1338    1760*   1791*   1819*   2985*   3001*   3012*   3044*   3059*
                         3088*   3135*   3203*   3592*   3606*   3616*   3709*   3728*   3741*   3754*   3767*   3780*   3793*
                         3806*   3819*   3970*   4026*   4037*   4048*   4143*   4161*   4185*   4218*   4249*   4272*   4291*
                         4955*   4977*   4999*   5021*   5043*   5065*   5087*   5109*   5131*   5153*   5175*   5197*   5663*
                         5716*   5759*   5802*   5848*
LOGICA  002560            645    1030#
LOKFLG  001312            716#
LOLIM   003434           1153*   1183    1198#
LPCNT   001224            677#   1075*   1076    1079*
LRC8  = 000200           5234    6092#
LSTERR  001234            681#    928*   1010*   1062*   1316    1318*   1402*
```

```
L.DATA  031754          4963    4985    5007    5029    5051    5073    5095    5117    5139    5161    5183    5205    5249
                        5294    5339    6040#
L00.03  001422           793#   1548*   1583
L04.07  001424           794#   1550*   1588
L08.11  001426           795#   1552*   1593
L12.15  001430           796#   1554*   1598
MAR17   014226          3136#
MASKX   001244           689#
MASK.A  001406           783#   1584
MASK.B  001410           784#   1589
MASK.C  001412           785#   1594
MASK.D  001414           786#   1599
MASTEK  005400          1340    1498#
MCRLF   005104          1109    1232    1336    1337    1345    1486    1498#   1607    1625
MCSRX   005330          1015    1498#
MDATA   005624          1259    1269    1511#
MEMEXT  014622          3210    3226    3236#   5944    5956
MEPASS  005145          1014    1498#
MERRPC  005454          1343    1498#
MERRX   005355          1021    1498#
MERR2   005174          1498#   1524    1714
MERR3   005243           969    1498#
MLOCK   005301           993    1498#
MNEW    005402           964    1498#
MPASSX  005344          1019    1498#
MPFAIL  005107          1399    1498#
MQM     005100          1139    1498#   1630
MR      005171          1001    1498#
MRESET= 004000           619#   1418    1431    3330
MSTCLR= 104412           749#   1403    3403    3439    3509    3535    3593    3647    3660    3672    3684    3710    3856
                        3914    3971    4096    4144    4162    4219    4241    4250    4273    4292    4321    4361    4432
                        4448    4460    4476    4488    4504    4516    4532    4544    4560    4572    4588    4600    4616
                        4628    4644    4670    4719    4834    4859    4874    4902    4956    4978    5000    5022    5044
                        5066    5088    5110    5132    5154    5176    5198    5239    5284    5329    5367    5407    5447
                        5490    5530    5555    5580    5605    5631    5664    5717    5760    5803    5849    5895    5943
MTITLE  001000           657#    946
MTSTN   005366          1341    1498#   1609
MTSTPC  005267          1498#
MVECX   005336          1017    1498#
NEXT    001216           674#   1082    1372    1759*   1790*   1858*   1884*   1910*   1936*   1962*   1990*   2021*   2047*
                        2073*   2099*   2125*   2151*   2179*   2208*   2232*   2253*   2281*   2309*   2337*   2365*   2393*
                        2421*   2447*   2473*   2499*   2525*   2551*   2577*   2603*   2630*   2667*   2707*   2730*   2756*
                        2782*   2808*   2834*   2860*   2886*   2912*   2942*   2984*   3043*   3087*   3202*   3254*   3320*
                        3402*   3438*   3508*   3534*   3591*   3646*   3708*   3855*   3913*   3969*   4095*   4139*   4320*
                        4360*   4428*   4669*   4718*   4833*   4901*   4950*   5233*   5278*   5323*   5365*   5405*   5445*
                        5488*   5528*   5553*   5578*   5603*   5630*   5662*   5715*   5758*   5801*   5847*   5894*   5942*
NOLIST= ****** U         539
NO.ATR  032232          5369    5386    5409    5426    5449    5467    5532    5557    5582    6104#
NO.BTR  032236          5370    5387    5410    5427    5450    5468    5533    5558    5583    5608    6108#
NPR   = 040000           614#   5729    5772    5820    5866    5905
NPRLOC  033352          5721    5726*   5734    5739*   5764    5769*   5777    5783*   5807    5812*   5815*   5823    5853
                        5858*   5861*   5869    6115#
PARAM = 104405           739#   1610
PARAM1  003304          1159#   1176
PARBIT= 040000           619#   1654
PARERR  003360          1162    1164    1166    1175#   1182    1184    1186
```

```
PASCNT  001230            679#   922*  1012*  1013   1046
PERFOR= 004537            619#  4399   4782   5681
PFTAB   004470           1400   1406#
POPR0 = 012600            593#  1366
POP1SP= 005726            593#
POP2SP= 022626            593#  1084
PS    = 177776            591#   918*   987*  1734*  3255*  3321*  5366*  5372*  5406*  5412*  5446*  5452*  5489*
                         5500*  5529*  5536*  5554*  5561*  5579*  5586*  5604*  5611*
PUSHR0= 010046            593#  1363
PUSH1S= 005746            593#
PUSH2S= 024646            593#
QV.FLG  001313            717#   924*  1025*  1073
RAM   = 020000            612#  4381   4397   4435   4463   4491   4519   4547   4575   4603   4631   4683   4724
                         4753   4906   4913   5692   5722   5765   5808   5816   5854   5862   5901   5964   5966
                         5969   5983   5985   5988   6008   6041   6046   6051
RAMCLR= 104413            751#  1404
RESREG  004300           1357   1360#
RESTAR  004414           1387   1393#
RESTRT  002572           1024   1028   1036#
RESV16  001404            777#  1573*  1574*  2731   2757   2783   2809   2835   2861   2887   2913   3134*  3153*
                         3166   3169*
RES05 = 104407            743#  1360
RETURN  001214            673#   930*  1000*  1002   1036*  1082*  1085   1372*  1374   1405   1624*  1632*  1633
ROMCLK= 104415            755#  3411   3600   3610   3621   3654   3666   3678   3690   3833   3861   3873   3884
                         3926   4003   4059   4061   4068   4070   4101   4107   4109   4114   4116   4149   4151
                         4172   4174   4176   4225   4227   4253   4255   4257   4259   4261   4276   4278   4280
                         4295   4297   4299   4326   4334   4336   4338   4385   4387   4404   4436   4438   4440
                         4464   4466   4468   4492   4494   4496   4520   4522   4524   4548   4550   4552   4576
                         4578   4580   4604   4606   4608   4632   4634   4636   4684   4736   4755   4757   4761
                         4777   4786   4839   4841   4847   4849   4851   4853   4862   4864   4866   4868   4877
                         4879   4881   4883   4885   4907   4909   4911   4914   4917   4919   4930   4932   4968
                         4970   4990   4992   5012   5014   5034   5036   5056   5058   5078   5080   5100   5102
                         5122   5124   5144   5146   5166   5168   5188   5190   5210   5212   5254   5256   5299
                         5301   5344   5346   5638   5672   5693   5723   5725   5730   5732   5766   5768   5773
                         5775   5809   5811   5817   5819   5821   5855   5857   5863   5865   5867   5902   5904
                         5906   5917   5960   5962   5965   5968   5970   5979   5981   5984   5987   5989   6009
                         6022   6025   6042   6044   6047   6049   6052
RUN     001304            707#   926*  1527   1530*  1531*  1538*  1539*
SAVACT  001302            705#   967   1712*
SAVNUM  001303            706#   921*  1023*  1026*  1705*
SAVPC   001276            702#  1207*  1378
SAVR0   001260            695#  1216*  1221   6127   6158
SAVR1   001262            696#  1215*  1222   6118   6129   6141   6166
SAVR2   001264            697#  1214*  1223   6120   6131   6151   6160
SAVR3   001266            698#  1213*  1224   6125   6138   6143
SAVR4   001270            699#  1212*  1225   6123   6136   6147   6155   6164
SAVR5   001272            700#  1211*  1226   6134   6145   6153   6162
SAVSP   001274            701#
SAV05 = 104406            741#  1320
SCOPE = 104400            729#  1774   1838   1872   1898   1924   1950   1976   2009   2035   2061   2087   2113
                         2139   2165   2198   2219   2241   2269   2297   2325   2353   2381   2409   2435   2461
                         2487   2513   2539   2565   2591   2617   2654   2692   2718   2744   2770   2796   2822
                         2848   2874   2900   2926   2972   3030   3073   3183   3234   3307   3372   3419   3497
                         3519   3575   3627   3696   3829   3899   3957   4055   4120   4306   4347   4407   4654
                         4705   4809   4890   4936   5219   5264   5309   5354   5390   5430   5471   5506   5539
                         5564   5589   5615   5649   5696   5741   5785   5831   5878   5925   6001
```

```
SCOP1 = 104401          731# 1768 1804 1826 2997 3007 3018 3065 3115 3222 3605 3615 3626
                       3659 3671 3683 3695 3724 3737 3750 3763 3776 3789 3802 3815 3828
                       4160 4184 4214 4245 4267 4286 4305 4976 4998 5020 5042 5064 5086
                       5108 5130 5152 5174 5196 5218 5680 5738 5782 5827 5874
SCREGO  031640         5957 5976 6005#
SERV.G  004640         1061 1308 1452# 1453
SETSCA  031676         4399 4782 5681 6016#
SETVEC  032210         5368 5378 5385 5408 5418 5425 5448 5459 5466 5493 5531 5556 5581
                       5606 6097#
SIMBCC  032030         4959 4981 5003 5025 5047 5069 5091 5113 5135 5157 5179 5201 5245
                       5290 5335 6056#
SKIP  = 000000         1841 1845 6113#
SPACNT= 003745         1242* 1263 1266* 1280#
SSWR    000176          650# 943
STACK = 001200          592# 919 988 1373 1395 5391 5431 5472
STAT    001236          686#
SV05    003452         1211#
SWR     001202          661# 938 943* 948 962 967 972 991 1055 1064 1071 1092 1105
                       1302 1309 1314 1361 1368 1370 1454 1455* 1465* 1470* 1471* 1472* 1479*
                       1602
SW00  = 000001          572# 962
SW01  = 000002          571# 1602
SW02  = 000004          570#
SW03  = 000010          569#
SW04  = 000020          568#
SW05  = 000040          567#
SW06  = 000100          566#
SW07  = 000200          565#
SW08  = 000400          564# 1368
SW09  = 001000          563# 1092
SW10  = 002000          562# 1370
SW11  = 004000          561# 1071
SW12  = 010000          560# 1105 1309
SW13  = 020000          559# 1314
SW14  = 040000          558#
SW15  = 100000          557#
SYNA00  001506          824#
SYNA01  001532          835#
SYNA02  001556          846#
SYNA03  001602          857#
SYNA04  001626          868#
SYNA05  001652          879#
SYNA06  001676          890#
SYNA07  001722          901#
SYNB00  001512          826#
SYNB01  001536          837#
SYNB02  001562          848#
SYNB03  001606          859#
SYNB04  001632          870#
SYNB05  001656          881#
SYNB06  001702          892#
SYNB07  001726          903#
SYNCX   001240          687#
SYNC00  001516          828#
SYNC01  001542          839#
SYNC02  001566          850#
```

```
SYNC03  001612          861#
SYNC04  001636          872#
SYNC05  001662          883#
SYNC06  001706          894#
SYNC07  001732          905#
SYNC2A  001432          798#  1549*
SYNC2B  001434          799#  1551*
SYNC2C  001436          800#  1553*
SYNC2D  001440          801#  1555*
SYND00  001522          830#
SYND01  001546          841#
SYND02  001572          852#
SYND03  001616          863#
SYND04  001642          874#
SYND05  001666          885#
SYND06  001712          896#
SYND07  001736          907#
S.C   = 050000          615#  3405  3597  3608  3618  3651  3663  3675  3687  3830  3860  3871  3883
                       3923  4056  4066  4277  4296  4333  4335  5915
TEMP    005562         1246  1396* 1397* 1509#
TEMP1   001246          690#   952*   953   958* 1500  1731* 1732* 3208* 3229* 4791* 4797  4806* 4807
                       5670* 5675  5683* 5686* 5690* 6059* 6082*
TEMP2   001250          691#   953*   954  1502  5667* 5689* 5690  5691  5694  6060* 6064*
TEMP3   001252          692#  6061* 6063  6073* 6075  6080* 6081* 6084
TEMP4   001254          693#  5946* 5994*
TEMP5   001256          694#  3210* 3213  3223* 3226*
TKCSR   001204          665#  1067  1126  1452  1459  1480                                        2
TKDBR   001206          666#  1057  1069  1128  1134  1304  1461  1482
TLAST = 031344         1628  6113#
TPCSR   001210          667#  1110  1132  1311  1466
TPDBR   001212          668#  1112* 1134* 1313* 1468*
TRPOK   003762         1292#
TSTNO   001226          678#   929* 1381  1408  1613  1620  1622  1758* 1789* 1857* 1883* 1909* 1935*
                       1961* 1989* 2020* 2046* 2072* 2098* 2124* 2150* 2178* 2207* 2231* 2252* 2280*
                       2308* 2336* 2364* 2392* 2420* 2446* 2472* 2498* 2524* 2550* 2576* 2602* 2629*
                       2666* 2706* 2729* 2755* 2781* 2807* 2833* 2859* 2885* 2911* 2941* 2983* 3042*
                       3085* 3201* 3253* 3319* 3401* 3437* 3507* 3533* 3590* 3645* 3707* 3854* 3912*
                       3968* 4094* 4138* 4319* 4359* 4427* 4668* 4717* 4832* 4900* 4949* 5232* 5277*
                       5322* 5364* 5404* 5444* 5487* 5527* 5552* 5577* 5602* 5629* 5661* 5714* 5757*
                       5800* 5846* 5893* 5941*
TST1    007260         1616  1632  1758#
TST10   010156         1962  1989#
TST100  021562         4360  4427#
TST101  023164         4428  4668#
TST102  023364         4669  4717#
TST103  024000         4718  4832#
TST104  024254         4833  4900#
TST105  024446         4901  4949#
TST106  026102         4950  5232#
TST107  026260         5233  5277#
TST11   010252         1990  2020#
TST110  026436         5278  5322#
TST111  026614         5323  5364#
TST112  026764         5365  5404#
TST113  027134         5405  5444#
TST114  027312         5445  5487#
```

```
TST115  027452            5488    5527#
TST116  027542            5528    5552#
TST117  027632            5553    5577#
TST12   010326            2021    2046#
TST120  027722            5578    5602#
TST121  030014            5603    5629#
TST122  030116            5630    5661#
TST123  030316            5662    5714#
TST124  030456            5715    5757#
TST125  030620            5758    5800#
TST126  031002            5801    5846#
TST127  031166            5847    5893#
TST13   010402            2047    2072#
TST130  031344            5894    5941#   6113
TST131= ****** U          5942
TST14   010456            2073    2098#
TST15   010532            2099    2124#
TST16   010606            2125    2150#
TST17   010662            2151    2178#
TST2    007370            1759    1789#
TST20   010756            2179    2207#
TST21   011034            2208    2231#
TST22   011074            2232    2252#
TST23   011160            2253    2280#
TST24   011244            2281    2308#
TST25   011330            2309    2336#
TST26   011414            2337    2364#
TST27   011500            2365    2392#
TST3    007622            1790    1857#
TST30   011564            2393    2420#
TST31   011640            2421    2446#
TST32   011714            2447    2472#
TST33   011770            2473    2498#
TST34   012044            2499    2524#
TST35   012120            2525    2550#
TST36   012174            2551    2576#
TST37   012250            2577    2602#
TST4    007676            1858    1883#
TST40   012324            2603    2629#
TST41   012426            2630    2666#
TST42   012536            2667    2706#
TST43   012600            2707    2729#
TST44   012654            2730    2755#
TST45   012730            2756    2781#
TST46   013004            2782    2807#
TST47   013060            2808    2833#
TST5    007752            1884    1909#
TST50   013134            2834    2859#
TST51   013210            2860    2885#
TST52   013264            2886    2911#
TST53   013340            2912    2941#
TST54   013474            2942    2983#
TST55   013676            2984    3042#
TST56   014036            3043    3085#
TST57   014434            3087    3201#
TST6    010026            1910    1935#
```

```
TST60   014630          3202    3253#
TST61   015044          3254    3319#
TST62   015262          3320    3401#
TST63   015364          3402    3437#
TST64   015630          3438    3507#
TST65   015706          3508    3533#
TST66   016074          3534    3590#
TST67   016266          3591    3645#
TST7    010102          1936    1961#
TST70   016510          3646    3707#
TST71   017172          3708    3854#
TST72   017414          3855    3912#
TST73   017620          3913    3968#
TST74   020160          3969    4094#
TST75   020300          4095    4138#
TST76   021210          4139    4319#
TST77   021336          4320    4359#
TTST    002702           994*    995*    997*    998*   1065#
TWOSYN= 010000           619#
TYPDAT  004266          1335    1353    1356#
TYPE  = 104402           733#     946     951     964     969     993    1001    1014    1015    1017    1019    1021    1109
                        1122    1139    1232    1269    1336    1337    1340    1341    1343    1345    1349    1354    1399
                        1456    1458    1486    1524    1607    1625    1630    1714
TYPMSG  004166          1333    1336#
VECMAP  007104          1713    1721#
WRDCNT  003742          1240*   1270*   1278#
WRKO.F  004254          1348    1351#
XBX     004060          1310    1312    1314#
XCSR    002604          1016    1038#
XERR    002626          1022    1047#
XFR   = 030000           613#    3997    4017    4028    4039    4050    4099    4147    4195    4226    4228    4254    4324
                        4383    4403    4437    4465    4493    4521    4549    4577    4605    4633    4726    4785    4837
                        4848    4863    4878    4890    4908    4910    4918    4931    4953    5235    5280    5325    5636
                        5671    5724    5731    5767    5774    5810    5818    5856    5864    5903    5958    5977    6043
                        6048
XHEAD   005461           951    1498#
XPASS   002620          1020    1044#
XPOLY   032202          4951*   5234*   5279*   5324*   6069    6080    6089#
XSTATQ  005506           957    1498#
XTSTN   004374          1342    1379#
XVEC    002612          1018    1041#
$CRAP = 177777             1#    1751#   1754#   1780#   1785#   1849#   1853#   1875#   1879#   1901#   1905#   1927#   1931#
                        1953#   1957#   1979#   1985#   2012#   2016#   2038#   2042#   2064#   2068#   2090#   2094#   2116#
                        2120#   2142#   2146#   2168#   2174#   2200#   2203#   2222#   2227#   2244#   2248#   2272#   2276#
                        2300#   2304#   2328#   2332#   2356#   2360#   2384#   2388#   2412#   2416#   2438#   2442#   2464#
                        2468#   2490#   2494#   2516#   2520#   2542#   2546#   2568#   2572#   2594#   2598#   2620#   2624#
                        2657#   2661#   2695#   2702#   2721#   2725#   2747#   2751#   2773#   2777#   2799#   2803#   2825#
                        2829#   2851#   2855#   2877#   2881#   2903#   2907#   2929#   2937#   2974#   2979#   3033#   3038#
                        3076#   3081#   3186#   3196#   3244#   3249#   3310#   3315#   3375#   3390#   3392#   3397#   3421#
                        3433#   3499#   3503#   3521#   3529#   3579#   3586#   3630#   3641#   3699#   3703#   3843#   3850#
                        3901#   3908#   3960#   3964#   4078#   4090#   4122#   4134#   4309#   4315#   4349#   4355#   4409#
                        4423#   4657#   4664#   4708#   4713#   4811#   4818#   4820#   4828#   4892#   4896#   4940#   4945#
                        5222#   5228#   5267#   5273#   5312#   5318#   5357#   5360#   5397#   5400#   5437#   5440#   5478#
                        5483#   5519#   5523#   5544#   5548#   5569#   5573#   5594#   5598#   5620#   5625#   5652#   5657#
                        5702#   5710#   5745#   5753#   5789#   5796#   5835#   5842#   5882#   5889#   5928#   5937#
$E    = 000132             1#    1759    1761#   1790    1792#   1858    1859#   1884    1885#   1910    1911#   1936    1937#
```

```
                    1962   1963#  1990   1991#  2021   2022#  2047   2048#  2073   2074#  2099   2100#  2125
                    2126#  2151   2152#  2179   2180#  2208   2209#  2232   2233#  2253   2254#  2281   2282#
                    2309   2310#  2337   2338#  2365   2366#  2393   2394#  2421   2422#  2447   2448#  2473
                    2474#  2499   2500#  2525   2526#  2551   2552#  2577   2578#  2603   2604#  2630   2631#
                    2667   2668#  2707   2708#  2730   2731#  2756   2757#  2782   2783#  2808   2809#  2834
                    2835#  2860   2861#  2886   2887#  2912   2913#  2942   2943#  2984   2986#  3043   3045#
                    3087   3089#  3202   3204#  3254   3255#  3320   3321#  3402   3403#  3438   3439#  3508
                    3509#  3534   3535#  3591   3593#  3646   3647#  3708   3710#  3855   3856#  3913   3914#
                    3969   3971#  4095   4096#  4139   4140#  4320   4321#  4360   4361#  4428   4429#  4669
                    4670#  4718   4719#  4833   4834#  4901   4902#  4950   4951#  5233   5234#  5278   5279#
                    5323   5324#  5365   5366#  5405   5406#  5445   5446#  5488   5489#  5528   5529#  5553
                    5554#  5578   5579#  5603   5604#  5630   5631#  5662   5664#  5715   5717#  5758   5760#
                    5801   5803#  5847   5849#  5894   5895#  5942   5943#
$N      = 000130      1#   1751   1756   1761#  1780   1787   1792#  1849   1855   1859#  1875   1881   1885#
                    1901   1907   1911#  1927   1933   1937#  1953   1959   1963#  1979   1987   1991#  2012
                    2018   2022#  2038   2044   2048#  2064   2070   2074#  2090   2096   2100#  2116   2122
                    2126#  2142   2148   2152#  2168   2176   2180#  2200   2205   2209#  2222   2229   2233#
                    2244   2250   2254#  2272   2278   2282#  2300   2306   2310#  2328   2334   2338#  2356
                    2362   2366#  2384   2390   2394#  2412   2418   2422#  2438   2444   2448#  2464   2470
                    2474#  2490   2496   2500#  2516   2522   2526#  2542   2548   2552#  2568   2574   2578#
                    2594   2600   2604#  2620   2627   2631#  2657   2664   2668#  2695   2704   2708#  2721
                    2727   2731#  2747   2753   2757#  2773   2779   2783#  2799   2805   2809#  2825   2831
                    2835#  2851   2857   2861#  2877   2883   2887#  2903   2909   2913#  2929   2939   2943#
                    2974   2981   2986#  3033   3040   3045#  3076   3083   3089#  3186   3199   3204#  3244
                    3251   3255#  3310   3317   3321#  3375   3392   3399   3403#  3421   3435   3439#  3499
                    3505   3509#  3521   3531   3535#  3579   3588   3593#  3630   3643   3647#  3699   3705
                    3710#  3843   3852   3856#  3901   3910   3914#  3960   3966   3971#  4078   4092   4096#
                    4122   4136   4140#  4309   4317   4321#  4349   4357   4361#  4409   4425   4429#  4657
                    4666   4670#  4708   4715   4719#  4811   4820   4830   4834#  4892   4898   4902#  4940
                    4947   4951#  5222   5230   5234#  5267   5275   5279#  5312   5320   5324#  5357   5362
                    5366#  5397   5402   5406#  5437   5442   5446#  5478   5485   5489#  5519   5525   5529#
                    5544   5550   5554#  5569   5575   5579#  5594   5600   5604#  5620   5627   5631#  5652
                    5659   5664#  5702   5712   5717#  5745   5755   5760#  5789   5798   5803#  5835   5844
                    5849#  5882   5891   5895#  5928   5939   5943#  6113#
$Y      = 000017      1#    720#   729    731#   733#   735#   737#   739#   741#   743#   745#   747#   749#
                    751#   753#   755#   757#   759#
.       = 033644    630#   631    634#   641#   642#   643#   644#   647#   649#   652#   656#   658#   703#
                    704#   705#   706#   707#   708#   819#   821#   822#   823#   824#   825#   826#   827#
                    828#   829#   830#   832#   833#   834#   835#   836#   837#   838#   839#   840#   841#
                    843#   844#   845#   846#   847#   848#   849#   850#   851#   852#   854#   855#   856#
                    857#   858#   859#   860#   861#   862#   863#   865#   866#   867#   868#   869#   870#
                    871#   872#   873#   874#   876#   877#   878#   879#   880#   881#   882#   883#   884#
                    885#   887#   888#   889#   890#   891#   892#   893#   894#   895#   896#   898#   899#
                    900#   901#   902#   903#   904#   905#   906#   907#   971    1060   1307   1389   1398
                   1412   1450#  1460   1467   1481   1508#  1510#  1512#  1526   1717   1738   3241#  4679
                   4745   4748   4771   5247#  5248#  5250#  5251#  5292#  5293#  5295#  5296#  5337#  5338#
                   5340#  5341#  5502   5674   5913   5923   6113#

.BEGIN  002332     987#
.CNVRT  003542     748   1233#
.CONVR  003536     746   1232#
.DATAC  004576     758   1438#
.DELAY  004476     754   1409#
.EOP    002436    1009#  5942
.ERRTA  033520    1328   6168#
.HLT    004002     637   1301#
.INSTE  003224     738   1139#
```

```
.INSTR  003120             736    1118#
.INST1  003140            1122#   1142
.MSG    003142            1120*   1123#
.MSTCL  004556             750    1430#
.PARAM  003244             740    1150#
.PFAIL  004402             635     920    1386#   1394
.RAMCL  004516             752    1417#
.RES05  003504             744    1221#
.ROMCL  004566             756    1434#
.SAV05  003444             742    1207#
.SCOPE  002634             730    1054#
.SCOP1  003020             732    1092#
.START  001742             653     918#    930
.TRPSR  003750             639    1289#
.TRPTA  001314             728#   1294
.TYPE   003044             734    1102#
```

```
DVEND      1#  1003
DVFRNT     1#   539
HLT      593#  1447  1776  1803  1825  1865  1871  1891  1897  1917  1923  1943  1949  1969  1975
        1996   2002  2008  2028  2034  2054  2060  2080  2086  2106  2112  2132  2138  2158  2164
        2185   2191  2197  2217  2240  2261  2268  2289  2296  2317  2324  2345  2352  2373  2380
        2401   2408  2428  2434  2454  2460  2480  2486  2506  2512  2532  2538  2558  2564  2584
        2590   2610  2616  2637  2643  2649  2675  2681  2687  2717  2737  2743  2763  2769  2789
        2795   2815  2821  2841  2847  2867  2873  2893  2899  2919  2925  2950  2956  2965  2971
        2996   3006  3017  3064  3114  3157  3160  3163  3168  3221  3271  3276  3282  3286  3290
        3296   3300  3306  3336  3341  3347  3351  3355  3361  3365  3371  3409  3415  3418  3451
        3460   3469  3478  3487  3496  3518  3547  3556  3565  3574  3604  3614  3625  3658  3670
        3682   3694  3840  3868  3882  3898  3922  3939  3950  3956  4006  4065  4074  4105  4112
        4119   4156  4166  4180  4194  4203  4211  4237  4244  4266  4285  4304  4330  4342  4391
        4447   4456  4475  4484  4503  4512  4531  4540  4559  4568  4587  4596  4615  4624  4643
        4652   4680  4693  4704  4746  4749  4772  4801  4845  4858  4872  4889  4928  4975  4997
        5019   5041  5063  5085  5107  5129  5151  5173  5195  5217  5261  5306  5351  5384  5394
        5424   5434  5465  5475  5505  5510  5515  5613  5634  5644  5648  5679  5737  5781  5826
        5873   5914  5924  5999  6105  6109
$AAA     538#  3444  3453  3462  3471  3480  3489  3540  3549  3558  3567
$ABREG   538#  4077
$ADDT1   535#  1751
$ADDT2   535#  1848  1874  1900  1926  1952  2011  2037  2063  2089  2115  2141  2243  2271  2299
        2327   2355  2383  2411  2437  2463  2489  2515  2541  2567  2593  2720  2746  2772  2798
        2824   2850  2876  2902
$ADDT3   535#  2221  2694
$ADDT4   535#  2619  2656
$ADDT5   535#  1779
$ADT6    535#  2167
$ADT6A   535#  1978
$ALURE   538#  3725  3738  3751  3764  3777  3790  3803  3816
$BINBC   538#  5221  5266  5311
$BUFFE     1#  1504
$BYTEO   538#  2928
$CHUCK   538#  4432  4460  4488  4516  4544  4572  4600  4628
$CK15      1#
$CK150     1#
$CLR.T     1#
$CRCX    538#  4955  4977  4999  5021  5043  5065  5087  5109  5131  5153  5175  5197
$CYCLE     1#  1513
$DATXF   538#  4022  4033  4044
$DOSFR   538#  3374
$EABIT   539#  5927
$EOP       1#  1003
$FINI      1#  6113
$GETFL     1#
$GETPA     1#  1608
$HEADE     1#   539
$HOHO1   538#  3599  3609  3620  3653  3665  3677  3689
$INIT1   535#  3243  3309
$INTA    538#  5356  5396  5436
$INTB    538#  5477
$MEMEX   535#  3185
$MSCAN   538#  4308
$MSG       1#  1498
$NSR1    538#  5619
$NSR2    538#  5701  5744  5788  5834
```

```
CZDVA-C MACY11 30A(1052)  03-OCT-79  14:07  PAGE 137
CZDVAC.P11    03-OCT-79 14:04          CROSS REFERENCE TABLE -- MACRO NAMES                    SEQ 0134

$NSR3      538#    5881
$PFAIL       1#    1382
$PRIO      538#    5518    5543    5568    5593
$RAMCL       1#    1409
$RXSHI       1#
$SCOPE       1#    1050
$SECT1     535#    2974
$SECT2     535#    3033
$SECT3     535#    3076
$SETCL     538#    3578
$SETLI       1#
$SETSC       1#    6015
$SETSY       1#
$SET.T       1#
$SILOI       1#
$SIMBC       1#    6056
$TRPDE       1#     729     731     733     735     737     739     741     743     745     747     749     751     753     755
           757
$TSTN        1#    1756    1787    1855    1881    1907    1933    1959    1987    2018    2044    2070    2096    2122    2148
          2176    2205    2229    2250    2278    2306    2334    2362    2390    2418    2444    2470    2496    2522    2548
          2574    2600    2627    2664    2704    2727    2753    2779    2805    2831    2857    2883    2909    2939    2981
          3040    3083    3199    3251    3317    3399    3435    3505    3531    3588    3643    3705    3852    3910    3966
          4092    4136    4317    4357    4425    4666    4715    4830    4898    4947    5230    5275    5320    5362    5402
          5442    5485    5525    5550    5575    5600    5627    5659    5712    5755    5798    5844    5891    5939
$TXSHI       1#
$VARIA       1#     655
$XFER      538#    3959
$XXCRC     538#    4939
$XZ          1#    1751    1754    1780    1785    1849    1853    1875    1879    1901    1905    1927    1931    1953    1957
          1979    1985    2012    2016    2038    2042    2064    2068    2090    2094    2116    2120    2142    2146    2168
          2174    2200    2203    2222    2227    2244    2248    2272    2276    2300    2304    2328    2332    2356    2360
          2384    2388    2412    2416    2438    2442    2464    2468    2490    2494    2516    2520    2542    2546    2568
          2572    2594    2598    2620    2624    2657    2661    2695    2702    2721    2725    2747    2751    2773    2777
          2799    2803    2825    2829    2851    2855    2877    2881    2903    2907    2929    2937    2974    2979    3033
          3038    3076    3081    3186    3196    3244    3249    3310    3315    3375    3390    3392    3397    3421    3433
          3499    3503    3521    3529    3579    3586    3630    3641    3699    3703    3843    3850    3901    3908    3960
          3964    4078    4090    4122    4134    4309    4315    4349    4355    4409    4423    4657    4664    4708    4713
          4811    4818    4820    4828    4892    4896    4940    4945    5222    5228    5267    5273    5312    5318    5357
          5360    5397    5400    5437    5440    5478    5483    5519    5523    5544    5548    5569    5573    5594    5598
          5620    5625    5652    5657    5702    5710    5745    5753    5789    5796    5835    5842    5882    5889    5928
          5937

. ABS.  033644     000

 ERRORS DETECTED:  0

 CZDVAC.BIN,CZDVAC.LST/CRF/SOL/NL:TOC=CZDVXX.SML,CZDVAC.P11
 RUN-TIME: 52 75 6 SECONDS
 RUN-TIME RATIO: 2158/134=16.0
 CORE USED:  34K  (67 PAGES)
```